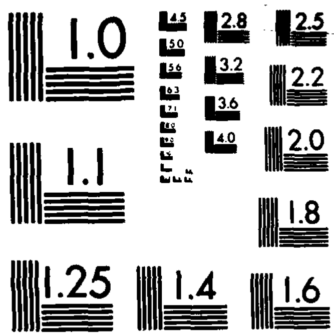
AD-A174 818 DESIGN NOTEBOOK FOR NAVAL AIR DEFENSE SIMULATION (NADS) 1/3
REVISION(U) TRW DEFENSE SYSTEMS GROUP MCLEAN VA
WATERWHEEL PROGRAM OFFICE 09 OCT 86 N00014-83-C-0527

UNCLASSIFIED F/G 15/3 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A174 818

DESIGN NOTEBOOK

FOR

NAVAL AIR DEFENSE SIMULATION (NADS)

Last Report:
October 15, 1982

Revised by:
Waterwheel Program Office
Defense Systems Group
TRW, 2751 Prosperity Ave., Fairfax,
9 October 1986

Prepared for
Office of the Chief of Naval Operations
OP-654

Contract N00014-83-C-0527

PREFACE

The Naval Defense Simulation (NADS) is a large scale simulation of the defenses of a Carrier Battle Group, CVBG, under attack by antiship missiles launched from ships, submarines, and bombers. NADS treats with considerable detail the airborne assets of the attacking Red force and the AAW assets of the Blue defending force. A prominent feature of NADS is its simulation of the Carrier Battle Group's acquisition of tactical information by its own resources, supplemented by external surveillance information. The battle group command center maintains a continually updated "Blue Perception" of the tactical situation. All of the tactical decisions are based on that perceived picture invoked by data that are often deficient in accuracy, completeness, and timeliness.

The NADS model is organized to fit the conventional defense-in-depth zones; the outer air battle; the surface-to-air, SAM, area defense; and the terminal defenses. It should be noted that a battle force without aircraft carriers is compatible with NADS, as any element of the CVBG may be omitted. Also the aircraft capability of a land airfield can be simulated with the appropriate inputs.

The Red attack force comprises bombers, escort fighters, recon aircraft, standoff jammers, and antiship missiles (ASM) that are launched by the bombers or by ships and submarines (the Red ships and submarines are not simulated). ASMs may be characterized by as many as five phases of flight including post-launch climb, high level cruise, mid-course descent, low level cruise and terminal maneuver. Missiles may have either nuclear or conventional warheads.

The simulation calculates detections for each BLUE unit and each RED unit based on the performance characteristics of the sensor carried by the unit, the size of the target, the radio horizon, and any sweep limits for the sensor. Self-screen electronic jamming and stand-off electronic jamming algorithms are also used to calculate a time of feasible detection. Actual detection is calculated using an empirical relationship to convert the predictions of the radar equation into a detection function which more closely agrees with real-world test data. Tracking delays and decision delays are also modeled.

The model treats jammers as being in the main-lobe depending on the geometry between the jammer, the victim and the target using an input main-lobe vulnerability to main lobe jamming. Provisions exist for both two-dimensional and three-dimensional main lobes selectable by input.

Algorithms calculate the time interval during which each jammer could be resolved by each victim in a multiple jammer enviroment. These calculations are used for both force and unit response to jamming.

When any unit succeeds in resolving a jammer in bearing, this result is reported to the command center. These data are maintained within the center and used, with appropriate errors, for generalized force tactics.

The model responds to jamming data by positioning fighter aircraft to cover the jammer. The data reported from units being jammed is used to guess a position for the jammer using the errored bearing from a unit with the least bearing uncertainty. If more than one unit in the force with the same bearing uncertainty has resolved the jammer, the range to the radio horizon of the furthest from the jammer position is used for the guess. This guess is then used to develop a bearing to the jammer from force center. A fighter is then positioned relative to this bearing. Angle off the bearing, stationing range and stationing altitude are input to the model for each type fighter.

Information that a jammer is close is an important consideration in both fighter stationing and fighter tactics. Two imprecise methods are used to cue fighters that jammers were close:

- o altitude ranging;
- o bearing ranging.

These methods are coded with the appropriate parameters in NADS.

The outer air battle is conducted by interceptors on combat air patrol (CAP) stations, deck-launched interceptors (DLI) from the carriers and fighters launched from land bases, using long range nuclear and conventional air-intercept missiles as well as close in missiles and guns. Coordinated by air controllers aboard ships or early warning aircraft which may be launched from either carriers or airfields, the interceptors normally are assigned targets by the command center but can select their own targets and engage if communications are disrupted by jamming or if the tactical situation requires urgent action. The objective is to intercept the bombers before the launch of anti-ship missiles.

Fighters are vectored toward a target by the controllers. When the fighter detects the target, the intercept is then controlled by the fighter. The fighter maneuvers to reach the Launch Acceptable Region, a volume of space around the target for each Air Intercept Missile dependent on the geometry of the intercept and the velocities of the fighter and the target. If the fighter can not achieve intercept for any reason: not enough fuel; intercept within the minimum intercept range; not enough speed; or target out of sensor field of view during the intercept, the LAR calculations are performed and if in LAR a missile is fired. Once in LAR, a weapon is fired and the results assessed. The engagement continues until either the target is destroyed or the fighter expends available fuel or weapons. If the fighter loses detection on the target, a request for controller support is made.

Fighter unit tactics provide for uncommitted fighter aircraft intercept of jammers when the jammer close cues were recognized. In the real world this would be accomplished by the fighter maneuvering to stop the bearing rate and then

closing with the target. Fighters can employ weapons against jamming aircraft using two modes: home-on-jam with data transfer from other units, and home-on-jam on jammer close cueing.

A capability to fire air-intercept-missiles (AIM) in home-on-jam mode using data transfer from other tracking units is incorporated. This mode uses an input launch acceptable region (LAR) reduction factor in calculating time to fire. If the reduction factor is greater than one, then the reduction factor actually expands the LAR. Provision for a different probability of kill when firing in this mode is furnished.

Fighters might fire AIMs when jammer close cues are developed. In this mode, the fighter uses own altitude and the type cue received, bearing rate or elevation angle, and an assumed range of one hundred n.mi. to perform a LAR calculation. If the mode is bearing rate, the fighter assumes that the target angle is ninety degrees. If the mode is elevation angle, the assumption is that the target angle is forty-five degrees. A worst case target altitude is assumed and the target is assumed to be at 600 knots. If the fighter is in LAR for any weapon carried, a missile is fired in the home-on-jam mode. If the fighter then burns through on the target, additional AIMs will be employed.

The interceptors will engage the Red missiles when feasible, but in most instances an antiship missile that is successfully launched will penetrate to the SAM area defense zone. This result is due to the small size of the Launch Acceptable Region for Air Intercept Missiles against anti-ship missiles in general and the limited number of firing opportunities usual in such an engagement. These conclusions derive from the input characteristics of the AIM's and the ASMs.

The Surface-to-Air-Missile (SAM) area defenses are conducted by anti-air-warfare (AAW) escort ships employing conventional or nuclear surface-to-air missiles. Three levels of coordination are selectable by input: coordination by the same command center that coordinates the outer air battle; coordination by sector assignment; and every ship for itself. The ships are characterized by their radar performance, number and type of missile launchers, number of fire control channels, number and types of missiles, and their tactical data net capability. The command center assigns specific targets to individual ships, as well as assigning sectors to be covered in the absence of target assignments. Each SAM firing ship has eight firing doctrines available by input for both primary and secondary employment doctrines. Each ship manages the commitment and tie-up times of its principal subsystems in the attempt to engage all targets. NADS currently has detection queues, fire control channel queues, launcher channel queues and illuminator queues associated with each SAM ship to provide for target prioritization and engagement sequencing with preemption. Targets that the SAM systems miss or cannot engage are passed on to the terminal defenses of the targeted ships.

Force and unit tactics which defer engagement to the dive portion of the target flight for very high altitude targets are integrated into NADS. The model has the capability to preempt system resources committed to an unengageable target in a very high altitude cruise in order to engage an engageable pop-up target.

The terminal defense phase is used in NADS to establish the number of hits for the case of Anti-Ship-Missiles (ASMs) with conventional warheads, or the time and position of the bursts for ASMs with nuclear warheads. The results for both hard kill devices, such as guns and point-defense missiles, and counter-measures are simulated.

Eight nuclear weapons enviroments are calculated for each nuclear burst, AIM, SAM, or ASM. Nuclear weapons effects are computed for each ship, aircraft, and missile in the proximity of each nuclear burst. Damage is scored by comparing the computed level with the input vulnerability threshold value for each type of potential victim with the nuclear enviroment created at the potential victim location.

Damage also results from the impact of conventionally armed weapons on a target. Damage is scored by cumulating the number of hits. Damage: up to six levels for ships; two levels for aircraft; and, three levels for missiles, caused during the battle on both sides reduces system capabilities for the remainder of the engagement.

The overall theme of the NADS simulation is to qualify the net performance of the defensive systems when their potential capabilities are limited by the imperfect tactical information. Because of incomplete and delayed information, in NADS as in actual combat, some targets are overengaged while others are unengaged, interceptors are placed on CAP stations too late or too soon, the stations are imperfectly placed, interceptors are sometimes assigned to fighters instead of bombers, and the launching of deck-launched interceptors (DLI) is not ideally timed. NADS was specifically designed to facilitate identification of the critical deficiencies and to evaluate the net worth of prospective ways to minimize their impact. The software has been designed in a modular form so that the effects of changes in hardware characteristics and in tactical logic can be readily examined without extensive reprogramming.

NADS was developed by the Waterwheel Program Office of TRW Defense Systems Group under contract to the Defense Nuclear Agency, and in coordination with the Office of the Chief of Naval Operations, OP-654. Continued development of the model has been sponsored by various government and corporate offices with OP-654 supervision.

A requirement of the NADS development program is the documentation of the simulation design and inputs. A prospective user can easily verify the algorithms that are used in the simulation using the documentation. This document is one of three that comprise the complete documentation of the NADS simulation:

- o NADS DESIGN NOTEBOOK

  A comprehensive description of the design of NADS, covering the tactical activities and physical phenomena that are being simulated and the design of the specific software elements that are used to perform the simulations. The design rationale and compromises are described, using the diagrams and flowcharts

that led to the coding of each module and subroutine.

- NADS SOURCE CODE

  The complete source code listings in GPSS and FORTRAN for all of the elements of NADS, extensively commented.

- NADS USERS' MANUAL

  User-oriented information on computer facility requirements, input data, input formats, user-constructed data files, and operating instructions to perform NADS simulations in a stand-alone mode or in concert with other models (e.g. NNWS).

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont'd)

TABLE OF CONTENTS (Cont'd)

TABLE OF CONTENTS (Cont'd)

CHAPTER 1

DESCRIPTIVE OVERVIEW

1.1 GENERAL

NADS is designed to apply the defense-in-depth concept providing multiple opportunities to intercept an attacking unit before it reaches the vital area (Battle Group center). It employs combat air patrol (CAP) aircraft, deck-launched interceptors (DLI), long range surface-to-air missiles (SAM), medium range SAMs and terminal defense weapons in subsequent order, to destroy the threat prior to impact in the vital area. The specific deployment of individual forces is variable to fit any particular combination of threat, available resources, mission, and geographical location.

The Red attack is composed of aircraft and antiship missiles. Four types of aircraft are modeled - bombers, reconnaissance, standoff jammers, and fighter escorts. Missiles may be air-launched, submarine-launched, or surface ship launched. Air-launched missiles are carried by bombers only. The submarine and surface ship missiles are treated jointly as sea-launched missile types.

The model is capable of accommodating both large and small scenarios. It can be easily resized, so there are few hard limits on the number of Red or Blue units.

1.2 BLUE FORCE COMPOSITION

The simulation incorporates a Carrier Battle Group (CVBG) consisting of aircraft carriers supported by multiple AAW and ASW ships. These ships along with aircraft from the carriers make up the AAW force. Operations of the ASW and surface warfare elements are not modeled explicitly in NADS, but non-AAW ships can be included as potential targets of the attack.

The major elements modeled are shown in Figure 1-1. Although the Officer in Tactical Command (OTC) is supported by coordinators responsible for the various specialized defense operations, they are not individually modeled in NADS. Figure 1-1 shows the primary elements that support the Antiair Warfare Coordinator (AAWC); the capabilities and responsibilities that are modeled for each are discussed in the following paragraphs.

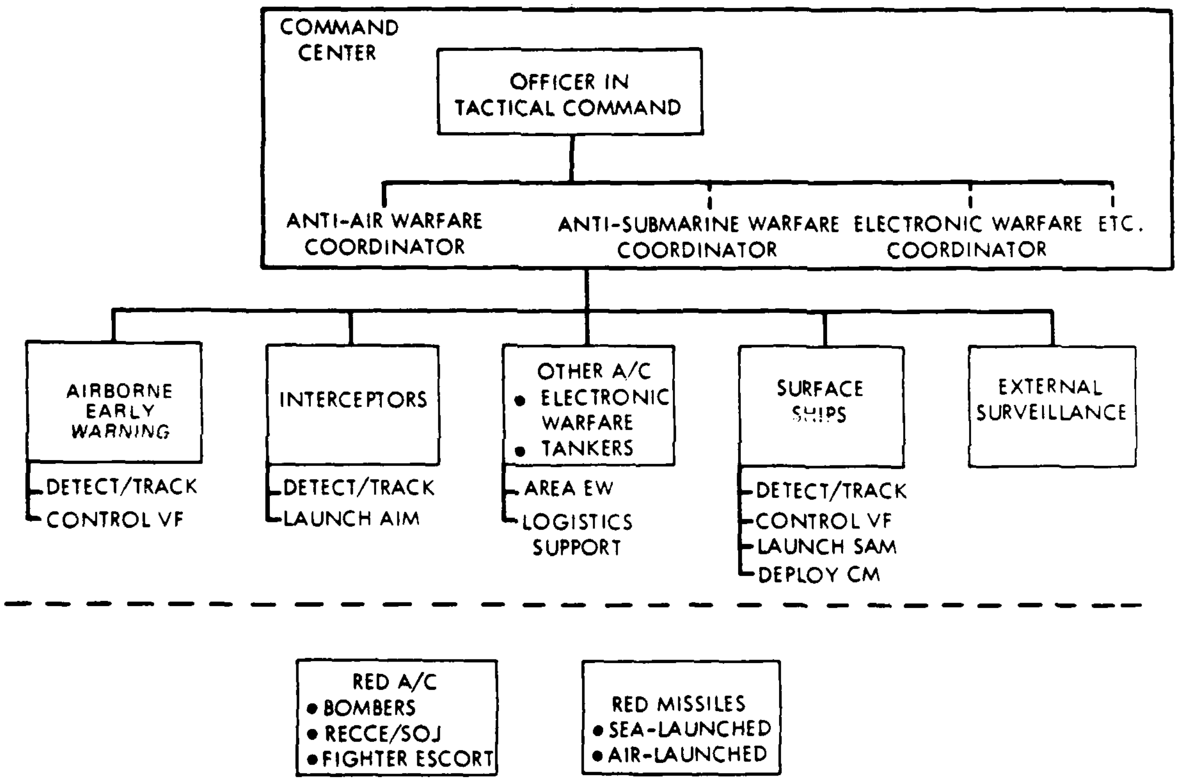


Figure 1-1. Major AAW Elements Modelled

1.2.1 Command Center

The CVBG command and control functions are aggregated into a single command center. The functions include those performed by the OTC, AAWC, and any interactions with other coordinators. The command center has access to information from messages, data networks, and external surveillance. It provides high level decision making and coordination among the elements of the defending force. A more detailed description of the command center function is given in Section 1.5, Command and Control.

1.2.2 Surface Ships

The surface ships are modeled to detect, track, and attack airborne threats. The number of launchers (one or two of the same type), target tracking and missile guidance capacity, and number and type (one nuclear-armed and one with conventional warhead) of missiles carried varies among ships. Therefore the simulation allows the user to select from ships of all major classes and to dispose of them in any desired information. SAM ships will generally be stationed near the vital area to be defended. Surface picket ships can be placed a large distance from the vital area to extend the range of detection and allow for early interception of threats. Ships have the ability to control airborne interceptors in attacks on enemy aircraft. The aircraft carriers, usually the protected units in the force, may also have SAM and terminal defense capability. NADS does not restrict the characteristics of ships to existing classes.

The ship model treats the following components:

- Air search radar;
- Radar tracking capacity;
- SAM fire control systems;
- SAM launching systems;
- SAM stores;
- Decision making;
- Defense countermeasures;
- Interceptor vectoring capability;
- Tactical data system; and,
- Communications.

The level of detail to which each component is modeled varies in proportion to the component's relative influence on the total ship peformance. Various ship types are accommodated by using different numeric values to characterize these components.

Detections are made on radars or passive detection systems. Radar detections develop target tracks that enable the decision maker to engage the track by use of SAMS, interceptors, countermeasures, or combinations of each. When a SAM launch is chosen, the target track is assigned to a fire control channel, and an appropriate SAM is selected for launching. After a SAM is launched, the fire control channel remains occupied until after the intercept. The ship will use data from the tactical data system and in turn contribute to it. The decision making function can also control interceptors and decide when and which countermeasures to use.

Ships are not moved during the simulation because of their low speeds, compared with aircraft and missiles, and the short duration of a typical AAW encounter.

1.2.2.1 Aircraft Carriers

A model of aircraft carrier operations is incorporated in NADS. This model permits flexdeck operations only; the model is not capable of cyclic operations. The primary purpose of the model is to furnish a realistic number of aircraft to the battle. The aircraft carrier module will attempt to maintain the alert posture stipulated in the input data. Force orders from the command center are processed to schedule the launch of aircraft. Aircraft return to the carrier after a mission and are recycled; a single aircraft may fly many times during a run of the simulation. The availability of aircraft is a primary constraint on operations as are the delays in launching caused by the launch equipment and the alert condition.

The aircraft carrier module performs the following functions:

- Launch of aircraft;
- Recovery of aircraft;
- Aircraft servicing;
- Aircraft repair;
- Maintenance of aircraft alert posture;
- Aircraft flight scheduling; and,
- Damage to launch equipment and embarked aircraft.

1.2.3 Airborne Early Warning

NADS provides for airborne early warning aircraft (VAW) which increase the detection range of the battle force by means of radar and passive detection, by being well forward of the main force and at altitude. The model contains the following components:

- o Surveillance radar;
- o Automatic and manual detect and track;
- o Air controller function;
- o Interceptor vectoring;
- o Tactical data sytstem; and,
- o Communications.

The level of detail to which each component is modeled varies in proportion to its relative contribution to the total early warning performance. Various aircraft types are accommodated by using different numeric values to characterize these components.

After detection, targets are tracked and the air controller function controls the assigned interceptors in their attacks on the targeted Red aircraft. The units may receive data from and transmit data to the tactical data net, and they serve as communications nodes between the command center and the interceptors.

The motion of the early warning aircraft is modeled. After takeoff from the aircraft carrier, the aircraft climbs to altitude using a normal climb schedule. The aircraft then proceeds toward a station using maximum range cruise. If the assigned station calls for bank-and-forth patrolling, the aircraft will fly out the northern edge of the station and then fly from edge to edge using an input patrolling speed. On other stations, maximum endurance fuel consumption is used and the aircraft motion is stopped. At the end of the mission (consumption of available fuel), the aircraft returns to the aircraft carrier, descends and lands.

1.2.4 Interceptors

Interceptors are fixed wing fighter aircraft (VF) capable of detecting and destroying airborne threats with air launched intercept missiles (AIM). Interceptors may be airborne on combat air patrol (CAP) stations or deck launched (DLI). The interceptor module contains the following components:

- o Fire control radar;

- o Target detection and tracking capability
- o Decision making;
- o AIM load; and,
- o Self vectoring capability

The level of detail to which each component is modeled varies with the component's relative influence on the total interceptor performance. The various interceptor types are accommodated by using different numeric values to characterize these components.

The interceptors are controlled from either ships or airborne controllers. Interceptors can also detect targets and carry out the final part of the attack on their own. The interceptor's decision logic must select among the AIMs available and decide when to launch them. An encounter with a Red fighter escort may occur if escorts are included in the Red flight plan. Time delays, weapon and fuel expenditure, and the outcome of such encounters are modeled.

The motion of the fighter aircraft is modeled. The interceptor model uses four aircraft speeds - loiter, cruise, normal intercept, and high speed intercept. Appropriate fuel consumption rates are used for each speed. After takeoff from the carrier when proceeding to a station in the fixed defensive posture or to a station to cover a reported raid, the aircraft climbs to altitude using a normal climb schedule. If called for in the inputs, the fighter will be fueled overhead the battle force prior to proceeding to a station in the fixed defensive posture. The aircraft then proceeds toward a station using maximum range cruise. If the assigned station calls for bank-and-forth patrolling, the aircraft will fly out the northern edge of the station and then fly from edge to edge using maximum range cruise speed. On other stations, maximum endurance fuel consumption is used and the aircraft motion is stopped. When prosecuting a target, the aircraft climbs to altitude using either a normal climb schedule or a mimimum-time-to-intercept climb schedule after takeoff. The aircraft then proceeds toward the target using the minimum speed necessary to achieve the intercept. At the end of the mission (consumption of available fuel or weapons), the aircraft returns to the aircraft carrier, descends and lands.

1.2.5 Electronic Support Aircraft

NADS provides for airborne electronic support (VAQ) which in future version will perform counter-targetting against the Red attack and passive detection of the Red attack. The model currently contains no components.

The level of detail to which each component will be modeled varies in proportion to its relative contribution to the total early warning performance. Various aircraft types are accommodated by using different numeric values to characterize these components.

The motion of the electronic support aircraft is modeled. After takeoff from the aircraft carrier, the aircraft climbs to altitude using a normal climb schedule. The aircraft then proceeds toward a station using maximum range cruise. On station, maximum endurance fuel consumption is used and the aircraft motion is stopped. At the end of the mission (consumption of available fuel), the aircraft returns to the aircraft carrier, descends and lands.

Other aircraft may be simulated, in future versions of NADS,

1.2.6 Other Aircraft

Other aircraft may be simulated, in future versions of NADS, to support the AAW mission in roles such as aerial refuelling. The principal effects of these aircraft will be modeled, but in less detail than the early warning aircraft and the interceptors.

1.2.7 External Surveillance

External surveillance as used here means any information on the size, composition, route, and timing of an attacking force gained from sources outside the CVBG itself. These sources, which may be satellites, other ships, or land-based sensors, are not individually modeled. The external surveillance is modeled implicitly through introduction of user-defined scenario-related surveillance cues. External surveillance cues are used to influence the decision making, such as surging of VF/VAW aircraft and selection of CAP and VAW station positions.

1.3 RED AIRCRAFT

The action of the threat aircraft is directed primarily by user-specified flight plans. Each waypoint of the flight plan is specified by coordinates at the beginning of each leg, and speed to the next waypoint.

1.3.1 Bombers

Bombers follow the present flight plan through most of the flight. The actual time of launching the ASCM can be a function of simulation conditions, such as defensive jamming. After launch of the last missile, two options are available: a simple escape course is set after an input post-launch air-to-missile guidance delay; otherwise, the flight plan continues to be executed.

Bombers can be assigned to carry up to four air-to-surface missiles (ASMs). The missile type and the target must be specified for each ASM in the scenario. Bombers may carry high power electronic transmitting equipment capable of jamming radars and communication channels for both self protection and for stand-off jamming.

1.3.2 Reconnaissance Aircraft

If the scenario includes reconnaissance aircraft, they will fly predefined flight plans only. They carry no missiles, but may use defensive self-screen jammers. Their presence may consume defense resources.

1.3.3 Stand-Off Jammer Aircraft

SOJ aircraft carry high power electronic transmitting equipment capable of jamming radars and communication channels from fairly distant positions. The jamming schedule and the intended targets of the jamming are part of the flight plan. Like the recon aircraft, the SOJ carries no weapons, and its entire flight plan is prespecified in the Red scenario.

1.3.4 Fighter Escorts

Fighters, at the scenario writer's option, may accompany the other three types of aircraft to provide protection against interceptors. The simulation does not include the details of dogfights between escorts and interceptors. If an interceptor is diverted by a fighter, the model causes the interceptor to expend fuel, weapons, and time, and it includes the possibility that the interceptor will be destroyed.

1.3.5 Use of Radar

The Red aircraft may use radar for reconnaissance or targeting. The detection of these transmissions can be utilized by the Blue force. On and off times can be randomized or controlled by a predefined schedule in the scenario. (The Red radar simulation is not implemented in the early versions of NADS.)

1.4 ANTISHIP MISSILES

The antiship cruise missiles (ASCM) are either air launched or sea launched. Air launched missiles enter the simulation at the bomber's coordinates at launch time. Since the launch point is variable in the simulation, the air launched missile flight plan is computed by NADS at the time of launch. The sea launched missile flight plans are computed during the initialization of the run, since the Red surface ships and submarines are modeled as fixed points with fixed launch schedules. The model can simulate a variety of flight profiles.

ASCMs are subjected to inflight failures, launch aborts and for low altitude flight, clobber. Launch conditions are checked for feasibility against the input missile characteristics.

1.5 DEFENSE STRUCTURE

The AAW defense structure is provided by the input scenario. Disposition of ships and early warning aircraft are fixed. An air plan specifies a fighter posture of fixed Combat Air Patrol (CAP) stations and an on-deck-alert schedule.

Fighters are also positioned in response to external surveillance reports. Prosecution of contacts by Battle Group sensors have highest priority followed by response to surveillance messages. Maintaining the input fighter posture is automatic when feasible.

The AAW assets are organized into a layered defense. This defense in depth is organized into three zones - forward air defense, SAM area defense, and terminal defense.

1.5.1 Forward Air Defense

The forward air defense is set up at distances from the force center that permit early detection of bombers with enough time to intercept them before ASCM missile launch. Although early warning aircraft or surface pickets are the primary means of initial detection in the forward air defense area, interceptors may also make initial detections. The interceptors are held on CAP stations until: (1) they are directed to a target by a controller, (2) they detect a target and attack on their own, or (3) they return to the carrier because of low fuel. The command decision logic compares the known target list with available interceptors and makes rational pairings. Assignments are not limited to one-on-one when the targets outnumber the interceptors.

An air controller normally vectors an interceptor from a CAP station until it has achieved its own detection. When there are not enough interceptors airborne, and the timing permits, deck launched interceptors will be launched. Nearest collision intercept vectors are used. The interceptor will normally detect the target during this process and prepare to complete the attack. To complete the attack the interceptor selects an appropriate weapon and launch strategy, approaches launch range, fires the weapon, assesses the result, and reattacks if required and if possible. Upon completing an attack, the interceptor will return to the carrier if its fuel or ammunition dictate. If not, it takes up a CAP station as its present position and is available for reassignment.

Air controller detections are shared with other platforms via the tactical data system(s). Tracks and responsibility for targets may be passed to other units if the controller's target handling capacity is exceeded. Surface pickets may also carry SAMs, which will be used if threats enter their envelopes.

1.5.2 SAM Area Defense

Threat aircraft or missiles that penetrate the forward air defense next encounter the SAM area defense, which is provided by a screen of guided missile ships around the vital area. The ships use air search radars for initial detection and identification of targets. Target tracks are transferred to fire control radars and their associated fire control computer channels, which determine target present and future positions, and compute launcher and missile prelaunch orders. Launching and handling systems prepare the SAM for firing. Single or salvo launches are selected, and after launching, the SAM is guided to its target. Methods of guidance vary somewhat from system to system and include

home-on-jam, home-all-the-way, mid-course guidance etc. NAD models these systems using delay times and intercept times. The endgame outcome is based on probabilistic data for the missile.

Assigned coverage sectors are defined around each ship. A ship may shoot at any target within its own sector. Depending on the current level of coordination control, ships may also fire into neighboring sectors as directed by the command center. Separation of CAP and SAM areas is defined by a user input range.

1.5.3 Terminal Defense

Those threats that penetrate the forward air defense and the SAM area defense are next engaged by the terminal defense systems of the targeted ships. The terminal defenses are modeled as an aggregate of short range missiles, guns, and electronic countermeasures. Separate off-line studies are relied upon to provide statistical definitions of the effectiveness of these systems for various classes of ships. An effective terminal defense is unlikely against nuclear missiles. The terminal defense phase is used in NADS to establish the number of hits by non-nuclear ASCMS and to establish the burst positions of nuclear ASCMs.

1.5.4 Area Electronic Warfare

Area electronic warfare uses jamming and deceptive countermeasures on an area wide basis. The primary platform is an EW aircraft orbiting over or near the Blue force center and controlled by the command center's decisions. (AREA electronic warfare is not implemented in the early versions of NADS.)

1.6 COMMAND AND CONTROL

A single command center is used in NADS to represent the Carrier Battle Group's OTC and other delegated AAW-related command functions, such as AAW coordination, EW coordination, and SAM area defense coordination. The command center formulates the required decisions from the information available to it, and implements those decisions by issuing command messages, force orders, to the relevant Blue units.

Three methods of area defense coordination are available by input:

- o Command center control;
- o Coordination by sector with command by exception; and,
- o General Melee with no coordination.

Mixing of these first two methods is the usual approach. In addition, when the ship module makes a determination based on input data that an ASCM is a threat to own ship, the ship will engage the target in self-defense.

In command center control, a target list, ordered by priority, is maintained for the battle force and each target is specifically assigned to a ship for engagement. There are three criterion for assignment. If the ship reports that it is engaging in self-defense, it has priority. If the target is currently engageable by any ship with capability, the target is assigned to the ship, of those which can engage, which can engage the target for the longest period of time. Otherwise, the target is assigned to the ship with capability which can engage the target first.

In sector coordination, a ship will engage targets in an assigned sector while notifying the command center. The command center will review the target list and cancel the engagement if another ship is already engaging the target. Preference is given to ships defending own ship. SAMs already launched will be guided to the target even if the command center issues a force order to cancel the engagement.

1.6.1 Available Information

The command center obtains its input data from four sources:

- The user-furnished scenario data;
- Input hardware characteristics; and,
- Task Force messages produced during the simulation run;
- Positional and velocity tracking data for Red units.

1.6.2 Scenario Data

The scenario specifies the initial conditions under which the simulation will be conducted. The initial conditions are defined by the following items, which represent prior decisions by the CVBG or higher commands:

- Nuclear weapons release status;
- Emissions release (EMREL) status;
- Positions of fighter stations;
- Positions of early warning aircraft stations;
- Positions of electronic support aircraft stations;
- Positions of ships;

- o Number of aircraft available on alert;
- o Limits of aircraft engagement zone; and
- o Assignment of SAM sectors to ships.

The CVBG command center model will simulate the receipt of any external surveillance information that the scenario may define, at the user's option. If the scenario specifies any such information, then each simulated surveillance message will contain at least the following data:

- o Time that the command center receives the message;
- o Time of the surveillance observation;
- o Position of the Red raid element observed.

In addition, the surveillance messages may be augmented to contain any or all of the following:

- o Uncertainty in the position datum;
- o Altitude of the raid element;
- o Number of aircraft;
- o Types of aircraft; and,
- o Mission or function of the raid.

1.6.3 Hardware Characteristics Data

The technical parameters and performance characteristics of the Blue units and their principal subsystem are provided to support the performance computations required by the simulation. The simulated command center has access to all such technical information on the Blue hardware.

The simulated command center is also provided some technical information on the Red aircraft for use in the decision making algorithm. A distinction is made between the Red data that the model user chooses to represent the actual performance of the Red systems, and and the data that represent the Blue intelligence view of the Red systems. The command center utilizes only the perceived and not the "actual" Red data.

For Red bombers, the command center is provided estimates (or data with which to formulate estimates) of: (1) missile launch range, and (2) probable attack / penetration speed.

The command center is furnished with:

- o Estimates of fighter kill potential against Red aircraft.
- o Estimates of early warning sweep widths.
- o Tactical objectives, such as keep out ranges.

1.6.4 Task Force Message Data

Except as precluded by battle damage or communications jamming, all status messages produced by Blue units in the course of the simulation are accessible to the command center. The Blue unit status messages include the following:

a. From all blue units:

Change in availability status of the unit.

Acquisition and loss of each radar track.

Recognition of Electronic Counter Measures (ECM).

Cessation of jamming.

b. From all fighters in flight:

Acceptance of target assignments.

Report target is enemy fighter.

Report of arrival on station.

Report of return to base.

Report if intercept is not possible.

Intercept result (kill or miss).

c. From air controllers (shipboard or airborne):

Acceptance of control for assigned aircraft.

Failure of air search radar.

Loss of tactical data system capability.

Loss of controller capability.

d. From ships:

Air search radar failures.

SAM system failures.

Loss of tactical data system capability.

Repair of failed radars.

Identification of targets being engaged.

Acceptance of target assignments.

Intercept results.

Cannot engage assigned target.

e. From aircraft carriers:

Aircraft status reports.

Acknowledgement of orders.

On-the-way reports.

1.6.5 Tracking Data

The command center uses positional and velocity information for Red aircraft and missiles being reported by the battle force. These data are used to prioritize the threats and determine engagement feasibility.

1.6.6 Decisions

Using appropriately selected portions of the body of available data defined in the foregoing paragraphs, the simulated command center makes the following decisions as they become relevant during the course of the simulation:

- o Designates targets to fighters;
- o Launch aircraft to fill specified CAP stations;
- o Launch fighters as DLI;
- o Position fighters to cover specified threats;
- o Position early warning aircraft to cover specified threats;
- o Position fighters in response to jamming;
- o Assign fighters to controller; and,
- o Designate SAM targets to ships.
- o Cancel target designations.
- o Order nuclear weapons employment.

In addition to the foregoing decisions, the model provides for evolutionary growth in capability for centralized combat coordination. The amount of detail in the status messages from ships to the command center is expandable by increasing the number of defined messages, or the contents of the messages, or both. The list of status arrays and decision making subroutines associated with the command center is expandable so that the additional status information can be used to develop and test methods for coordinating intership operations to minimize the number of unengageable or surviving targets.

1.6.7 Communications

The quality and timeliness of tactical decisions at the individual unit level and at the force level depend heavily on the performance of the communication links. NADS does not simulate the communication links, per se, but the delay times associated with the transfer of essential information are modeled. The receipt of each of the messages is represented by user-selectable delay times, and the delays are subject to further degradation by communications jamming and by battle damage. The delay times represent the collective influence of several factors -- the time to comprehend a new situation and initiate the appropriate message, the transmission time through the communication link, and the time to comprehend the message content and initiate a response.

1.6.8 Sensors

Radars are the primary means of detecting the threat, and detection range is the most critical characteristic that is modeled. Factors which may vary within an encounter and cause the detection range to change greatly are part of the model. These factors include horizon effects, target size, counter-measures and counter-counter-measures. Search and early warning radars are represented by full 360-degree coverage except where degraded by jamming. A moving fighter sweeps out only a sector.

A simplistic visual detection capability is modeled for fighters. This capability is characterized by a parameter in the model.

All units identify and report ECM when a jammer can be resolved and when the jamming stops. In addition, bearing rates of the resolved jammer and elevation angles to the jamming aircraft are calculated for fighter aircraft.

1.7 DEFENSIVE WEAPONS

Two principal categories of defensive weapons used are air-to-air missiles and surface-to-air missiles. Air-to-air missiles are launched from interceptors. Surface-to-air missiles (SAM) are launched from ships. An additional category is the point defense weapons of individual ships, comprising short range missiles and guns.

1.7.1 Air-to-Air Missiles

Four types of airborne intercept missiles (AIM) can be available concurrently on any interceptor. The first type is a long-range, nuclear tipped, active or semi-active missile. The second type has long range performance with a maximum of sophistication and capability. This type can be launched in salvos or one at a time and has either active or semiactive radar guidance. The third type is a shorter range AIM to be used as a backup to the primary weapon, and the fourth is a short range weapon, typified by Sidewinder. Hardware details are not modeled; system performance is characterized as a function of target geometry, firing positions, and other factors that may limit or greatly affect performance.

The model provides for mixed loads. The missile count on each aircraft is maintained throughout a flight as a factor in decision making. Actual track of the AIM flight is not simulated, but time and position of impact or miss is computed. Determination of a hit or miss is made at the time of launch, and is based on known (or user postulated) effectiveness data for the AIM.

1.7.2 Surface-to-Air Missiles

A variety of SAM types that differ in range capability, guidance, and warhead are modeled. Control delays, illuminator and fire control channel limitations, as well as launching and handling system delays are included in the model. The effects of target size, altitude, and speed are considered through the use of

variable missile coverage envelopes. The actual track of the SAM flight is not simulated, but time and position of intercept is computed. Determination of hit or miss is made at that time for particular type of SAM. The missile count for each SAM type in each ship is maintained throughout a replication for decision making and for calculating expenditures.

1.7.3 Terminal Defense Systems

Terminal (or point) defense systems consist of short range surface-to-air missiles, guns, and associated weapon handling and fire control systems. They are not individually modeled, but use effectiveness data from other sources to represent the entire system for each particular class of ship.

1.8 WEAPONS EFFECTS

1.8.1 Non-nuclear Weapons

The effects of a non-nuclear defense weapon against an attacking aircraft or missile are scored as a miss or a hit. The effects of non-nuclear attacking (Red) missiles against the ships of the CVBG are scored as misses or hits, and the damage level on each ship is estimated from the number of hits scored during the simulation run.

1.8.2 Nuclear Weapons

The simulation computes the time and position of each nuclear burst. The input data define the warhead characteristics and the hardness thresholds for each attacking unit and for each defending unit and its major subsystems. Separate thresholds can be specified for damage due to each of eight nuclear environments that are modeled, and for several levels of damage. For each nuclear burst, the simulation examines every Red unit and every Blue unit to determine whether it was within the damage envelopes.

The damage to units and subsystems is applied to the defense capabilities for the remainder of the simulation run. The cumulative effects at the end of the run can be printed out as one of the principal results of the engagement.

CHAPTER 2

SOFTWARE DESIGN APPROACH

The approach used in designing the software for the NADS Model is described in the following sections. The topics discussed include programming languages, data considerations, general software structure, and event scheduling.

2.1 PROGRAMMING LANGUAGES

NADS uses a combination of two programming languages, GPSS (General Purpose Simulation System) and FORTRAN. Each language has strengths which are exploited in NADS.

2.1.1 GPSS

GPSS is a widely available and popular computer language used to model complex systems consisting of many inter-related elements. The language is highly macro in character and provides a tool that enables many types of event-stepped models to be built quickly. GPSS is available on IBM 360 and 370 computers, on the VAX 11/780 series computers using VMS operating system, on the VAX 8600 with VMS, on the Amdahl 470V/6, the PDP-10, the Burroughs 5700 and 6700, the Univac 1108, and the CDC 6000 series, among others. The language can also be used on the time sharing networks of Computer Science Corporation, National CSS, ADA-Cyphernetics, and University Computing Corporation.

A GPSS program consists of a sequence of GPSS statements, called "blocks". There are more than forty Blocks which GPSS makes available to the model builder. It is the interaction among the Blocks which is analogous to (simulates) the interaction of elements in the real system being modeled. The action defined in a block takes place when a "transaction" moves through it.

Transactions are GPSS entities that are defined by a model builder to represent one or more elements of the situation being modeled. They usually represent some moving element, such as a customer moving through a checkout line, or a missile entering an air defense system.

In NADS, GPSS is used for transaction generation; control of logical event sequencing by transaction movement through the system; event scheduling; control of the simulation clock; probability distribution functions; and collection of statistical data. GPSS is used for queueing in the Detection and Tracking Module and in the SAM Ship Module. Most of the Aircraft Carrier Module is coded in this language.

2.1.2 FORTRAN

The FORTRAN programming language was selected because it is a widely used computer language that closely resembles the language of mathematics and was designed primarily for scientific and engineering applications. FORTRAN compilers are available for nearly all of the computers used most often by industry, government, and colleges and universities. One of the principal advantages of FORTRAN is that it provides the analyst with an efficient means of writing computer programs requiring after a relatively short period of instruction.

FORTRAN subroutines are used for functions for which GPSS is not well suited. These include reading input data; performing numerical computations; and maintaining large data files containing status data. At the current time two hundred eleven subroutines have been created for NADS.

The Programming standards and style guide for NADS requires that the FORTRAN be structured. This requirement is intended to produce readible and maintainable computer code.

2.2 DATA CONSIDERATIONS

A considerable quantity and variety of data is required for the Model. The data is organized into a hierarchical data structure. These data fall into two major categories; scenario data and technical data. Both types of data are maintained as disk files, which can be accessed during a simulation run.

The scenario data define the initial battle conditions and the prescheduled events for a simulation run. Included in the scenario are the disposition of the Blue forces; the Red attack plan; the surveillance messages to be received by Blue from sources outside the game; and definition of other "rules of the game" which have been left to the discretion of the model user.

The second category of data, called technical data, consists primarily of the physical characteristics of various types of hardware. These data are considered to be more static than the scenario data. However, even these values will change occasionally when hardware variations are being studied, or when more accurate data are obtained for previously defined hardware. Maintaining the data in a disk file rather than as hard-coded program constants facilitates such changes.

To reduce the possibility of error in the data base, formated data files are used for the input data. The software reads input data in convenient human units and converts to program variables in internal (meters, kilograms, seconds) units.

As these data are read into the internal data structure of the model, consistency checking is performed and the data written into the output report for review.

2.3 GENERAL SOFTWARE STRUCTURE

The general structure of the NADS software is presented in Figure 2-1. The preparation of the scenario and technical data files is shown above the dashed line, as taking place offline to the actual simulation. Below the dashed line are shown the GPSS and FORTRAN programs and the associated inputs and outputs that comprise the simulation software.

The simulation processing is controlled by the GPSS main program. The principal interface between GPSS and FORTRAN is through a single driver routine called HLPRTN. Two FORTRAN subroutines, INIT and APINIT, are used to load the required scenario and technical data from disk and perform conversions. The GPSS then calls other FORTRAN subroutines to perform computations as needed, and to produce any special formatted reports needed from the run. The GPSS program maintains the various event chains that control the sequence and timing of simulation events and accumulates extensive statistics that can be printed at the end of a run.

It should be noted that the scenario and technical input data are loaded into two types of internal data storage, GPSS savevalues and FORTRAN Common. When the model is installed on an IBM computer, however, an address conversion must be performed to enable access to the savevalues. This address conversion is required whenever a GPSS savevalue serves as input or output of a FORTRAN computation and also at startup time when savevalues are initialized with scenario input data (which are read in by the FORTRAN subroutines INIT and APINIT). The need for address conversion is minimized by keeping as much of the model as feasible in FORTRAN except where GPSS has a clear advantage. Additionally, the interface between the FORTRAN and GPSS is limited to HLPRTN, subroutines in the initialization section of the model and to subroutine RANODE in the Red Threat Module.

2.4 EVENT SCHEDULING

There are two approaches to event step modeling:

- o Schedule events and test just before execution to determine whether or not the event is still valid. Execute only valid events.

- o When a new event is determined, find the last tentative event scheduled and modify it to be the new event to be scheduled at the proper time. Execute all events as they occur.

The first approach can lead to large event lists with many events never being executed. The second approach uses computer resources to search the event list for the one to modify. The trade-off is between CPU time and computer storage. NADS takes the second approach.

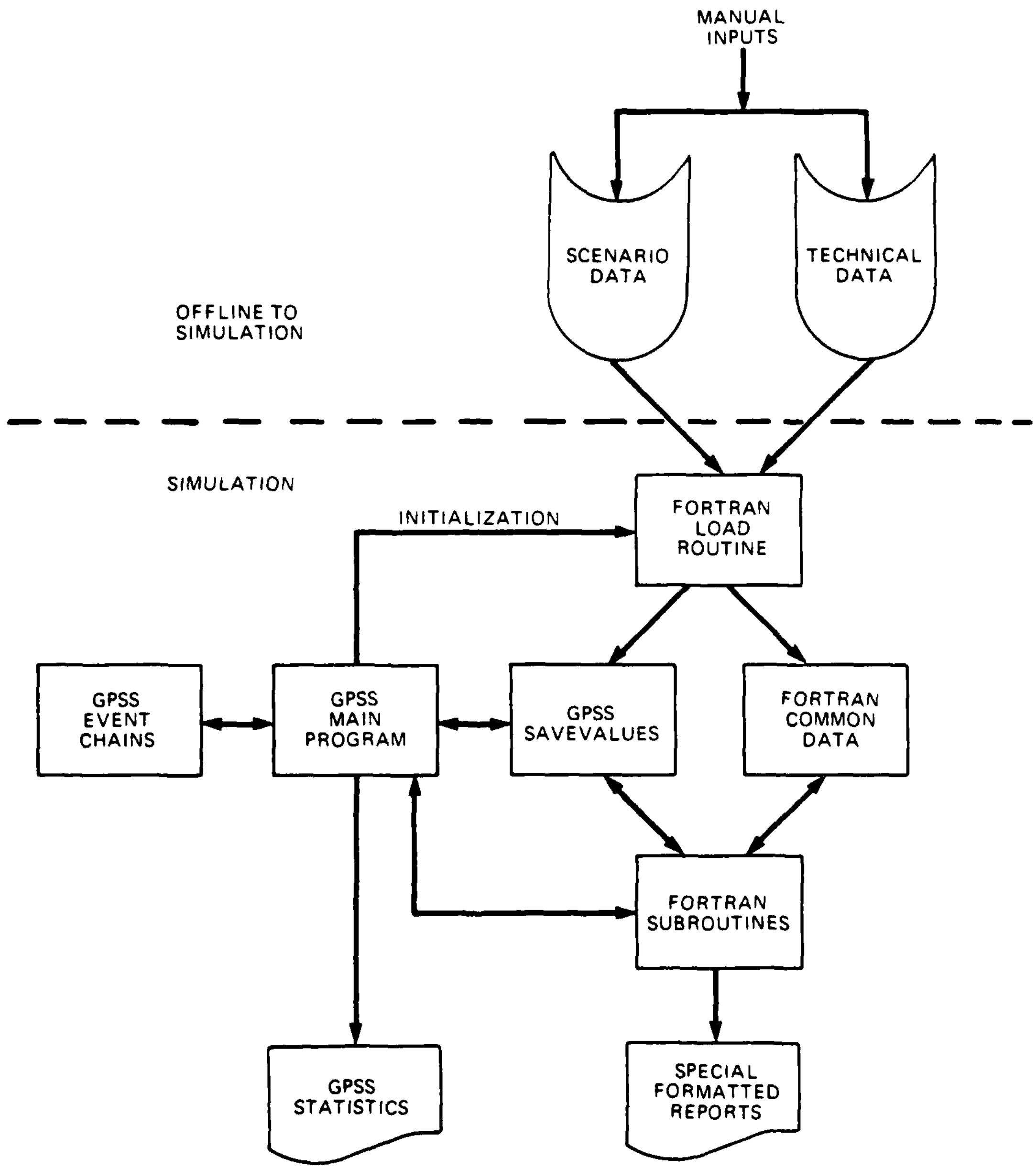


Figure 2-1. General Software Structure

GPSS makes use of several transaction chains (ordered lists of transactions) in the management of simulation events. The two primary ones are the current events chain (CEC) and the future events chain (FEC). These chains are maintained by the internal GPSS software. Transactions residing on these chains are not directly accessible by a GPSS user program.

In addition, GPSS provides for the creation of one or more user chains by a GPSS program. The program has complete control over its user chains, with the ability to link and unlink transactions as needed.

Table 2-1 summarizes the four principal transaction chains used by NADS.

The NADS software makes extensive use of a user chain that is designed to hold tentatively scheduled events. The storage of transactions representing future events on the user chain under program control, rather than relinquishing control to GPSS, enables the program to cancel or reschedule the events based on the occurrence of other prior events. Tentatively scheduled events are maintained on the user chain, sequenced by event time. This provides maximum access to tentatively scheduled events to enable an event to be rescheduled right up to the time of the event.

Since GPSS savevalues are used to communicate between transactions, the sequence of transaction processing can be critical. The number of transactions on the current events chain; i.e., the transactions in the model at any time is limited. Transactions are removed from the user chains one at a time when the associated event is to occur. Only after one transaction has been processed (moved through as many GPSS blocks as possible at the current clock time), and all other transactions unlinked by this transaction have been processed, is the next user chain transaction unlinked. Transaction priority is used to sequence the processing when transactions are unlinked by a transaction in the model.

Table 2-1. Summary of Principal GPSS Chains as Used in NADS

CHAIN	PURPOSE	SEQUENCE**	CONDITIONS FOR TRANSACTION TO MOVE ONTO CHAIN	CONDITIONS FOR TRANSACTIONS TO BE REMOVED FROM CHAIN
USER CHAIN 1 (UC1)	User defined. In NADS, used as list of tentatively scheduled events. Transactions can be selectively accessed by a GPSS program to reschedule events based on the occurrence of other events.	User defined. In NADS, ordered by tentative event time* (*Transaction parameter assigned by program).	User controlled by moving the transaction into a GPSS LINK block. In NADS this occurs when the initial or current processing of a transaction is completed and the tentative time of the next event has been computed. The transaction is moved from the CEC to the UC.	User controlled. Transactions are selectively moved from the UC to the CEC by another transaction's movement into a GPSS UNLINK block. In NADS, transactions are removed one at a time from the front of the chain as part of the event step process. Transactions are removed from any part of the chain in order to terminate them or recompute the event
USER CHAIN 2 (UC2)	Temporary storage for sorting & for detection computation at current time.	User defined	User controlled by GPSS link block. In NADS, whenever detection must be recomputed.	User controlled by GPSS unlink block. In NADS, last thing done in current time when event in future is next on UC1.
FUTURE EVENTS CHAIN (FEC)	Interal GPSS chain of definite future events. Not accessible by program.	By priority level within event time* (*As defined by a GPSS ADVANCE or GENERATE block)	GPSS automatically puts the next transaction to be generated by each GENERATE block on the FEC. Transactions are moved from the CEC to the FEC when they move into a positive ADVANCE bllock. The only ADVANCE blocks in NADS are in the CV module and in the DRIVER following the unlinking of the next event.	GPSS automatically moves transactions from the FEC to the GEC when the simulation clock is advanced to the event time for the transaction.

Table 2-1. Summary of Principal GPSS Chains as Used in NADS (Cont.)

CHAIN	PURPOSE	SEQUENCE**	CONDITIONS FOR TRANSACTION TO MOVE ONTO CHAIN	CONDITIONS FOR TRANSACTIONS TO BE REMOVED FROM CHAIN
CURRENT EVENTS CHAIN (CEC)	Internal GPSS chain of transactions to be processed at the current clock time. Not accessible by program.	By priority level	GPSS automatically moves transactions from the FEC to the CEC when the Clock is *advanced* to the transaction's event time. Transactions are moved from the UC to the CEC when unlinked by another transaction. Copy transactions are put on the CEC when created by the movement of their Parent transaction into a SPLIT block.	GPSS automatically moves transactions from the CEC to the FEC when they move into a positive ADVANCE block. Transactions are moved from the CEC to the UC when they move into a LINK block. Transactions are removed from the CEC (to the GPSS Latent Pool) when they moved into a TERMINATE block.

**In all chains, in case of ties, transactions will be in the order they were linked to the chain within the defined sequence.

CHAPTER 3

SIMULATION DESIGN

This section describes the basic design of the simulation in terms of software organization into modules, logical processing flow, and data structure.

Figure 3-1 provides an overview of the NADS software. The software functions are organized into several modules (depicted as blocks in the diagram) which simulate the capabilities and actions of the element that were described in Section 1. These modules are organized into three groups -- simulation control, Red attack, and Blue defense -- which are described below.

3.1 SIMULATION CONTROL

The Driver Module, which does not correspond to any real world elements, provides program control under GPSS. It generates and processes a Control transaction and a Timer transaction that initializes and closes out, respectively, the processing of a simulation run. The Control transaction also controls the event-stepping process.

As described in Section 2, transactions representing future events are stored on the user chain. Each transaction will have parameters associated with it that define the event time and the address or location of the GPSS code that will simulate the event. The Control transaction unlinks transactions (one at a time) from the front of the user chain when the next event is to occur. The unlinked transaction causes the simulation clock to be advanced to the event time and then transfers to the event address for processing by the appropriate module.

Note that transactions may be linked to the user chain from any part of the model, whenever the next event associated with a transaction is determined and the event time is computed. However, they will always be UNLINKED by the Control transaction whenever the occurrence of the event is to be simulated unless the event happens before the last transaction UNLINKED.

Transactions are unlinked in other sections of the model only when events are to be rescheduled or cancelled. For example, the detection of a Red aircraft by a ship may require rescheduling if the Red aircraft changes course. If the Red aircraft is shot down by a fighter prior to the detection by the ship, the detection event will be cancelled. To facilitate this type of processing, each transaction on the user chain has parameters assigned that indicate the next scheduled event and provide for the recomputation of event times and for terminating the transaction (thus cancelling all future events for it).

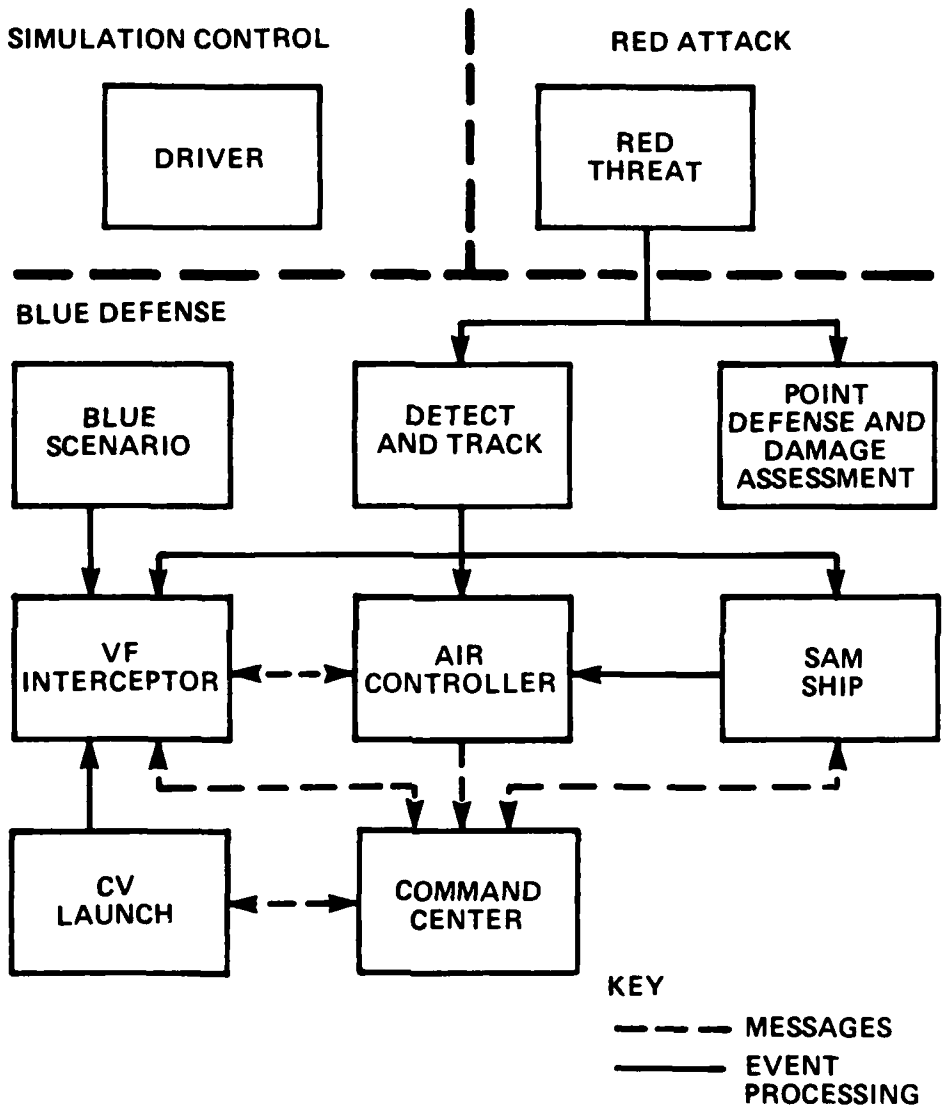


Figure 3-1. Organization of NADS Software Modules

3.2 RED ATTACK

The Red Threat module generates and processes transactions representing the aircraft and missiles comprising the Red attack. One transaction, called a Red parent transaction, will be generated for each Red aircraft, each surface-to-surface missile (SSM) and each subsurface-to-surface missile (SSM) defined in the input scenario. Air-to-surface missiles (ASM) will also be represented by Red parent transactions when they are launched. The term "parent" distinguishes these transactions from the sensor "copy" transactions that are generated for each Blue unit to represent interactions between Red and Blue.

The Red parent transactions are used to simulate events associated with the various nodes of an aircraft or missile flight path. A transaction is terminated if the aircraft or missile is shot down. Otherwise, the last event for an aircraft will be to leave the simulation (terminate) when it leaves the battle area to return to its base. The last event for a missile will be the warhead burst. When this event occurs, the Damage Assessment module will be executed to determine the effects.

Copies of each Red parent transaction are generated for each Blue unit and sent to the Detect and Track module. These sensor copy transactions are used to simulate the detection of the Red target and the defensive actions taken. Whenever a Red parent event occurs (such as a course change), the events scheduled for that parent's sensor copy transactions (such as detection by a ship) are re-evaluated.

3.3 BLUE DEFENSE

Most of the NADS software is devoted to the simulation of defensive actions taken by the Blue forces. Since the actions of the Blue units will be almost entirely related to the various Red threat elements, these actions are simulated using the sensor copy transaction. As indicated above, one copy of each Red parent transaction is generated for each Blue unit and sent to the Detect and Track module for processing.

The Detect and Track module simulates the detection and tracking functions of all Blue units, based on the detection range and tracking capacity of the individual units.

When the Detect and Track module determines that a Red target is being tracked by a Blue unit, the sensor copy transaction representing that Red threat-Blue unit combination is transferred to the appropriate module for decision or action as follows:

Type unit tracking target	Decision/Action Module
Early Warning Aircraft	VAW and Air Controller
Fighter Aircraft	Interceptor
Ship	SAM Ship or Air Controller

The Command Center module is notified of all tracks to simulate both NTDS and manual reporting functions.

The Air Controller module simulates the control of all types of aircraft and the vectoring of fighters. Both early warning aircraft and ships which have fighters under their control are modeled in this module.

The SAM Ship module simulates the SAM area defense against all Red targets that get through the forward air defense.

The Terminal Defense and Damage Assessment module simulates the use of defensive countermeasures and short range weapons when Red targets get through the SAM area defense, and assesses the damage caused by conventional and nuclear missiles.

The Interceptor module simulates the decision and actions of fighters. While in holding, the fighters are treated as if they were stationary. When patrolling a station, on intercept, enroute to a station or returning to base, movement is simulated. Fighters may be vectored to a target by an air controller or they may make their own detections and go after their own targets.

The VAW module simulates the flight decisions and actions of early warning aircraft. While in holding, the aircraft are treated as if they were stationary. When patrolling a station, enroute to a station or returning to base, movement is simulated.

The VAQ module simulates the flight decisions and actions of electronic support aircraft. While in holding, the aircraft are treated as if they were stationary. When enroute to a station or returning to base, movement is simulated.

The Command Center module simulates the various centralized command functions. The launching of DLI and CAP from carriers is performed in the CV module.

Communications between the various Blue units are simulated via message transactions that are sent between modules to trigger a decision and actions. Delays between the time the message is sent and the time it is received (if at all) are based on the type of communcation link involved and any communication jamming* in effect.

The Blue Scenario module initiates other events related to the Blue forces that are independent of the Red threat actions. These include scheduling events for fighters that are airborne at simulation start; surveillance messages that are received from sources outside the game; and equipment failures.** Transactions generated to simulate events of these types are based on the input scenario.

* Communications jamming is not currently implemented in NADS.

3.4 GPSS TRANSACTIONS

Several types of transactions are used for scheduling and performing the functions described below. Table 3-1 summarizes the transaction types that occur in each module and describes how each type is used.

Table 3-2. GPSS Transaction Summary

MODULE	TRANSACTION TYPE	PROCESSING
Driver	Control	GPSS and FORTRAN initialized. Controls event stepping for the simulation
Driver	Timer	At the scheduled end of the simulation, initiates the generation of special reports and terminates the process.
Red Threat	Red Parents (Aircraft and Missiles)	Controls flight path, radar, jamming, and missile launches of Red aircraft and flight of Red missiles according to the input scenario and computations.

** Blue hardware malfunctions are not currently implemented in NADS.

Table 3-1 continued. GPSS Transaction Summary

MODULE	TRANSACTION TYPE	PROCESSING
Red Threat	Active Sensor copies (Aircraft and Missiles)	Sensor copies of Red Parent transactions are generated for each Blue sensor and sent to the Detect and Track module. These transactions are used to represent the knowledge and defensive actions of each Blue Unit against individual Red units. The sensor copy transaction is also used as the main action copy; i.e., the future event for the pair is on this transaction.
Red Threat	Passive Sensor copies (Aircraft)	Passive Sensor copies of Red aircraft Parent transactions are generated for each Blue sensor. These transactions are used to represent knowledge of Red jamming and to stimulate fighter responses.
Red Threat	Messages (pseudo)	Pseudo messages are sent to the Command Center when Red units are killed or depart the simulation and when course changes occur. These simulate loss of track events as well as changes to target tracks.
Blue Scenario	Aircraft Control	Transactions are generated at the beginning of each simulation for each aircraft. These transactions are used to schedule aircraft carrier events as well as events not associated with a specific Red, Blue pair such as arrival on station, arrival at the turn point when patrolling the station and return to the carrier at the end of a mission.
Blue Scenario	External Surveillance	This transaction is used to schedule the receipt of reports by surveillance systems exogenous to the battle force. The transaction also stimulates the Command Center response.

Table 3-1 continued. GPSS Transaction Summary

MODULE	TRANSACTION TYPE	PROCESSING
Blue Scenario	Command Center Control	This transaction is used to stimulate Command Center decision either at the end of current time or at some future time determined by the FORTRAN COMAND Subroutine.
Detect and Track	Active Sensor	This transaction is used to store the time of detection, the time of lost detection and the current detection status for a Red, Blue pair. Upon initiation of tracking, this transaction is sent to the module corresponding to the Blue unit making the detection. Until the time of lost detection, the transaction is used to schedule the next event for the pair.
Detect and Track	Passive Sensor	This transaction is used to store the time when a Red unit which is radar jamming will be resolved by a Blue unit, the time when jamming ceases or the jammer is no longer resolved and the current status with the jammer.
Detect and Track	Messages (Reports)	Messages are sent to the Command Center whenever tracking is initiated for a new target by any Blue unit, or when contact with the target is lost. Reports are made when a jammer is first resolved and again when jamming is no longer observed.
Detect and Track	Messages (Perception)	Messages are sent to the Fighter Module whenever a determination is made that a jammer has approached the fighter within a distance suitable for engagement.
Detect and Track	Messages (Perception)	Messages are sent to the Red threat Module to signal start track. With suitable delays, these perceptions stimulate reactive jamming.

Table 3-1 continued. GPSS Transaction Summary

MODULE	TRANSACTION TYPE	PROCESSING
Command Center Module	Messages (Orders and Reports)	Messages will be received from the Detect and Track Module regarding new Red targets being tracked or recognition of jamming activities (represents messages from individual Blue units). Messages exchanged with Air Controllers regarding interceptor assignments. Messages exchanged with SAM ships regarding target assignments. Messages exchanged with the Aircraft Carrier Module regarding the launch of aircraft as Deck Launched Interceptors of Combat Air Patrol. Messages are received from all Blue units regarding operational status.
Command Center Module	External Surveillance Message	Messages regarding enemy sightings are used to update data arrays representing knowledge of enemy positions. Depending on the tactical situation, decisions to launch additional fighters or early warning aircraft or to reposition airborne aircraft may be made.
Air Controller (VAW and Ship)	Active Sensor Message	Assigns and vectors fighters to intercept targets being tracked.
Air Controller (VAW and Ship)	Messages (Orders and Reports)	Messages are sent between the controller and the fighters regarding fuel state, assignments, outcomes of engagements, vectoring, self-assignment confirmation and status. Messages are sent between the controller and the command center regarding fighter status, intercept and target assignments.

Table 3-1 continued. GPSS Transaction Summary

MODULE	TRANSACTION TYPE	PROCESSING
Interceptor	Active Sensor	Fighters initiate tracking of Red units in the detect and track module. Tactical decisions are made which cause action events to be scheduled on this transaction for this Red, Blue pair. Target priority is also maintained.
Interceptor	Aircraft Control	Use to schedule future events that are not associated with a specific Red, Blue pair such as arrival at station, arrival at the turn point when patrolling the station and return to the carrier at the end of a mission.
Interceptor (VAW and Ship)	Message (Orders and Reports)	Messages are sent between the Air Controller and the fighter regarding status, assignments, vectoring and outcomes of engagements.
Interceptor Track	Message (Perception)	Messages are sent from the Detect and Track Module when a Red jammer closes the fighter. This transaction can cause self-assignments and AIM firing.
Interceptor	AIM	This transaction is created when an air intercept missile is fired. The Red, AIM outcome is scheduled.
SAM Ship	Active Sensor	Ship defense events are scheduled by this transaction. Additionally, engagement status and target priority are maintained for the Red, Blue pair.
SAM Ship	SAM	This transaction is created when a Surface-to-Air Missile is fired. The Red, SAM outcome is scheduled.
SAM Ship	Message (Orders and Reports)	Messages are sent between the Ship and the Command Center regarding target assignments and operational status.

Table 3-1 continued. GPSS Transaction Summary

MODULE	TRANSACTION TYPE	PROCESSING
Terminal Defense Damage Assessment	Active Sensor	Processed to resolve the future event for Red, Blue tactical pairings.
Terminal Defense Damage Assessment	Damage	Imposes the damage calculated at the appropriate time.
Terminal Defense Damage Assessment	Red Parent	Processed to resolve the future event for damaged Red units.
Terminal Defense Damage Assessment	Message	Messages are sent to the Command Center, via the air controller for aircraft, for status reporting.

Table 3-2 defines the contents of the transaction parameters.

Table 3-2. GPSS Transaction Paremeters

P1 - Red Target ID
(Aircraft Control) - Aircraft Type with Red ID
P2 - Time Next Event
P3 - Event Address
P4 - Event or Message Type
P5 - (Sensor) - Time of lost detection
(Aircraft Control) - Station assignment
(Red parent/damage) - Cause of nuclear burst
P6 - (Active Sensor) - Latest time to wait for SAM resource
(Passive Sensor) - Method used
(SAM) - Time detection lost
(Aircraft Control) - Initial fuel state
P7 - Blue Unit ID (Primary) (Making detection, subject of status message, having failure, etc.)
P8 - Transaction type:
1 = RED parent
2 = Active Sensor Copy
3 = Passive Sensor Copy
13 = SAM Copy
20 = AIM Copy
20 = Message
21 = External Surveillance Message
30 = Control
31 = Timer
32 = Blue Failure
33 = Aircraft Control
34 = SAM Reload
35 = Blue Damage
36 = Red Damage
37 = Command Center Control
P9 - (Sensor) - Current State Detection/Tracking
-1 = Undetected
0 = Undetected
1 = Detected - in Tracking Queue
2 = Tracking
(Passive Sensor) - Current State Detection
0 = Undetected
1 = Detected
2 = Jammer is close
(Message) - Blue ID Sending Message
(Aircraft Control) - CV number and alert state
(SAM reload) - number of launcher to reload

Table 3-2. GPSS Transaction Paremeters continued

P10 - (Message) - Blue ID Receiving Message
(Sensor) - fighter Being Vectored by Controller in P1
(Aircraft Control) - Initial Weapon load
P11 - (Sensor) - SAM Status (SMSTAT)
0 = Not Processed
1 = Target Engageable
2 = In Queue for FCC
3 = Have FCC - Looking On
4 = In Queue for Launcher Channel
5 = 1 Launcher Channel - In Slew
6 = 2 Launcher Channels - In Slew
7 = In Illuminator Queue - 1 launcher
8 = In Illuminator Queue - 2 launchers
9 = In Illuminator Queue - vertical launcher
10 = Ready to fire - one launcher
11 = Ready to fire - two launchers
12 = Ready to fire - vertical launchers
13 = Firing, Past Point of No Return (1 LC)
14 = Firing, Past Point of No Return (2 LC)
15 = Firing, Vertical Launcher
20 = Not Engageable, or Cancelled by CC
21 = Not in Sector
22 = Engaged by Another Ship
23 = Waiting for Response to Self-Assign NUC Request
(Message) - Auxiliary Message Data (Counts,
CAP Station No. for New Launch,
External Surveillance Message Number)
P12 - (Sensor) - Target Priority (PRIORITY)
0 = Not a Target for this BUID
1 = Threat to Own Ship, Nuclear SAM
2 = Threat to Own Ship
3 = CC Assignment, Nuclear SAM
4 = Command Center Assignment
5 = Self Selected, Unconfirmed
6 = Threat to Own Ship, Nuclear SAM, above SAM ceiling
7 = Threat to Own Ship, above SAM ceiling
8 = CC Assignment, Nuclear SAM, above SAM ceiling
9 = Command Center Assignment, above SAM ceiling
10 = Self Selected, Unconfirmed, above SAM ceiling
(Aircraft Control) - Initial Weapon Load
(Message) - CAP assignment for aircraft
P13 - (Sensor) - Sensor type number of Blue unit
(Red Parent) - Jammer Looping parameter
(Aircraft Control) - Initial Weapon Load
P14 - Identifier of the user chain to which to link
the transaction.

3.4.1 Message Transaction

Message transactions use parameter P4 to define the message content. The code carried on P4 is defined by the following tables. The messages are categorized by their principal subject matter, with the hundreds digit of P4 indicating the category:

001 to 009 - General Purpose (multiple platform and weapons types)
101 to 199 - Aircraft Employment
201 to 299 - SAM Employment
301 to 399 - Blue Unit and System Status

The message list is given first in Table 3-3 as brief mnemonic labels for convenience in diagrams and commented listings, and then in Table 3-4 with specific definitions.

Table 3-5 displays the transaction parameters for each message type.

* (Set negative when a 2nd detection is moved to P2 as the next event. DETECT must be called again to compute a possible 3rd detection)

Table 3-3. Message Types (P4 Code and Mnemonic)

General Purpose

001 Ext Surv
*002 Detection
003 Lost Detection
004 Tracking
*005 Track Assign
006 Target Accepted
007 Target Rejected
008 Hit Target
009 Missed Target
010 Red Change

Aircraft Employment

101 Target Assign
102 New Vector
103 Self Assign
104 Self Assign OK
105 Self Assign NoGo
106 On Self Control
107 Red Fighter
108 Intercept Handover
109 Accept Handover
110 Immediate Launch Order
111 VF Engaging Tgt
112 Reject Handover
113 Can't Comply
114 Cancel Launch Order
115 Revise Launch Order
*116 Salvo Fighter
117 Deferred Launch Order

SAM Employment

201 Target Assign
202 Assign Nuc
203 Self Assign
204 Self Assign Nuc
205 Self Assign Nogo
206 Self Assign Notice

Blue Status

301 Assign Controller
302 Accept Control
303 Unit Down
304 On Cap
*305 TDS Down
306 Controller Status Change
307 Reject Control
*308 Voice Jam
309 Radar Change
*310 FC Tracker Change
311 FC Channel Change
*312 Launcher Change
313 SAMS
314 Nuc SAMs
315 SAM Local
316 Enroute to Station
317 Go On Cap
318 CV Aircraft Status Report
319 Enroute to CAP
320 BINGO (Return to Base)

*Not Currently Implemented.

Table 3-4. Message Definitions

Message transaction numbered 001 represents the external surveillance message whose serial number is carried in P11. Read the message from the Blue Scenario using P11 as the index.

Message transaction numbered 002 represents the Blue Unit, P7, and has radar contact on Red Unit, P1.

Message transaction numbered 003 represents the Blue Unit, P7, and has lost radar contact with Red Unit, P1.

Message transaction numbered 004 represents the Blue Unit, P7, and is tracking Red Unit, P1.

Message transaction numbered 005 represents the Blue Unit, P7, and is assigned to maintain track on Red Unit, P1, and provides principal input to the Tactical Data Net.

Message transaction numbered 00 represents the Blue Unit, P7, acknowledges its assignment to intercept Red Unit, P1, and will attempt to do so.

Message transaction numbered 007 *represents the* Blue Unit, P7, is unable to intercept Red Unit, P1, and therefore rejects assignment to it.

Message transaction numbered 008 represents the Blue Unit, P7, has fired on Red Unit, P1, and hit it.

Message transaction numbered 009 represents the Blue Unit, P7, and has attempted to intercept Red Unit, P1, but failed to hit it. Thus, P1 is a leaker.

Message transaction numbered 010 represents the Red Unit, P1, flight plan change. Message not jammable.

Message transaction numbered 101,represents Blue Unit, P7 (a VF) is assigned to intercept Red Unit, P1, (at course and speed provided in FORTRAN Common) and *attack it with AIMS.*

Message transaction numbered 102 represents the Blue Unit, (P1).

Message transaction numbered 103 represents the Blue Unit, P7, intends to assign itself to engage Red Unit, P1, with AIMS.

Message transaction numbered 104 represents the Blue Unit's, P7, self-assignment to Red Unit, P1, is approved.

Message transaction numbered 105 represents the Blue Unit's P7, self-assignment to Red Unit, P1, is overruled.

Message transaction numbered 106 represents the Blue Unit, P7, is changing state from controlled intercept of Red Unit, P1, to self-vectored intercept of P1.

Message transaction numbered 107 represents the Blue Unit, P7, is reporting that Red Unit, P1,is a fighter.

Message transaction numbered 108 represents the Blue Unit, P10, is assigned as controller of Blue Unit, P7, (a VF), and P7's previously assigned intercept of Red Unit, P1.

Message transaction numbered 109, represents Blue Unit, P9, accepts control of Blue Unit, P7, and of its previously assigned intercept of Red Unit, P1.

Message transaction numbered 110 represents the VF to be launched immediately from CV. Red Unit assignment is in P1. CAP station number can be in P11.

Message numbered 111 represents the Blue Unit, P7 (a VF), is engaging Red Unit, P1.

Message numbered 112 represents the Blue Unit, P9, rejects control of Blue Unit, P7, and of its previously assigned intercept of Red Unit, P1.

Message numbered 113 represents the Blue Unit, P7 (a CV), and cannot comply with launch order.

Message numbered 114 represents the Command Center order to Blue Unit (a CV) to cancel launch.

Message numbered 115 represents the Command Center order to Blue Unit (a CV) to revise a deferred launch time.

Message numbered 116 is not currently implemented.

Message numbered 117 represents the Command Center order to Blue Unit (a CV) to schedule the launch of an aircraft.

Message numbered 201 represents the Blue Unit, P7, is assigned to Red Unit, P1, with non-nuclear SAM.

Message numbered 202 represents the Blue Unit, P7, is assigned to engage Red Unit, P1,with nuclear SAM.

Message numbered 203 represents the Blue Unit, P7, is assigning itself to engage Red Unit, P1, with non-nuclear SAM.

Message numbered 204 represents the Blue Unit, P7, is assigning itself to engage Red Unit, P1, with nuclear SAM.

Message numbered 205 represents the Blue Unit's P7, self-assignment to Red Unit, P1, is overruled.

Message numbered 206 represents the Blue Unit, P7, will engage Red Unit, P1, which is threat to own ship.

Message numbered 301 represents the Blue Unit, P10, is assigned as controller of Blue Unit, P7 (a VF).

Message numbered 302 represents the Blue Unit, P9, accepts control of Blue Unit, P7 (a VF).

Message numbered 303 represents the Blue Unit, P7, is Down, i.e., it cannot perform any useful part of its normal mission.

Message numbered 304 represents the Blue Unit, P7 (a VF), is changing state to "On CAP Station". P1 = Red Unit if it was assigned.

Message numbered 305 represents the Blue Unit's, P7, Tactical Data System capability is Down.

Message numbered 306 represents the Blue Unit's, P7, Air Controller vector capability is changed. (0 = Down; 1 = Standard; 2 = Can Accept More).

Message numbered 307 represents the Blue Unit, P9, rejects control of Blue Unit, P7 (a VF).

Message numbered 308 represents the Blue Unit's P7, Voice Link reception is jammed, and some messages or all messages to P7 may be lost.

Message numbered 309 represents the number of air search radars that Blue Unit, P7, has operating is changed to the number in P11.

Message numbered 311 represents the number of Fire Control channels operable in Blue Unit, P7, is changed to the number in P11.

Messge numbered 313 represents P11 which is the number of useable conventional SAMs remaining in Blue Unit, P7.

Message numbered 314 represents P11 and is the number of useable nuclear SAMs available in Blue Unit, P7.

Message numbered 315 represents the Blue Unit, P7, is ordered to local control of target selection from any Red Unit in its envelope.

Message numbered 316 represents the Blue Unit, P7, is VF which was launched. (P1 = 0 if CAP; P1 = Red ID of target if DLI).

Messge numbered 317 represents the Blue Unit, P7 (a VF), is ordered to halt attack and take up CAP Station at current location.

Message numbered 318 represents the report from P7 (a CV) on the status of on-deck aircraft.

Message numbered 319 represents the Blue Unit, P7 (an aircraft) is enroute to station.

Message numbered 320 represents the Blue Unit, P7 (an aircraft) is returning to BASE.

Table 3-5. Parameters for Message Transactions (PB2=20)

PH4 MESSAGE TYPE	MESSAGE		PB1 BLUE UNIT ID	PH1 RED UNIT ID	PB3 BLUE ID SENDING MSG.	PB4 BLUE ID RECEIVING MSG.	PB5 AUXILIARY MSG. DATA
001	Ext Surv		-	-	-	CC	Msg. Serial No.
002							
003	Lost Detection		Ship, VAW, VF	Target	PB1	CC	-
004	Tracking		" " "	"	"	"	-
005							
006	Target Accepted	(1)	VF	"	"	Air Contrl	-
		(2)	Ship, Air Contrl	"	"	CC	-
007	Target Rejected	(1)	VF	"	"	Air Contrl	0=Self Vector
		(2)	Ship, Air Contrl	"	"	CC	1=AC Vector
008	Target Hit/Left	(1)	VF	"	"	Air Contrl	-
	Game	(2)	Ship, Air Contrl	"	"	CC	-
009	Missed Target	(1)	VF	"	"	Air Contrl	-
		(2)	Ship, Air Contrl	"	"	CC	-
010	Track Change		-	"	0	"	-
101	Target Assign	(1)	VF	"	CC	Air Contrl	-
		(2)	VF	"	Air Contrl	PB1	-
102	New Vector		VF	"	"	"	-

Table 3-5. Parameters for Message Transactions (PB2=20) (Cont.)

PH4 MESSAGE TYPE	MESSAGE		PB1 BLUE UNIT ID	PH1 RED UNIT ID	PB3 BLUE ID SENDING MSG.	PB4 BLUE ID RECEIVING MSG.	PB5 AUXILIARY MSG. DATA
301	Assign Controller		VF	-	CC	Air Contrl	-
302	Accept Control		"	-	Air Contrl	CC	-
303	Unit Down	(1)	"	-	PB1	Air Contrl	-
		(2)	Ship, Air Contrl	-	"	CC	-
304	On CAP	(1)	VF	Target (if any)	"	Air Contrl	-
		(2)	Air Contrl	" "	"	CC	-
305							
306	Controller Status Change		Air Contrl	-	PB1	CC	0 =Down 1=Saturated 2=Can accept more
307	Reject Control		VF	0	Air Contrl	"	
309	Radar Change		Ship	-	PB1	"	No. Radars Left
310							
311	FC Channel Change		"	-	"	"	No. Channels Left
312							
313	SAMs		"	-	"	"	No. Conv. SAMs
314	Nuc SAMs		"	-	"	"	No. Nuc SAMs
315	SAM Local		"	-	CC	PB1	-
316	Enroute to Station		VF	0 if CAP Target if DLI	CV	CC	Station if CAP 0 if DLI
317	Go on CAP		"	-	CC	VF	-
318*	CV Aircraft Status Report		CV	-	CV	CC	-
319	Enroute to CAP		VF	Target	CC	Air Control	Air Control VF

Footnote:
* PB3 - CV Number; PBS - Aircraft Type
PH6 - Number in alert state 2; PH5 - Number in alert state 1;
PB6 - Number in alert state 3.

Table 3-5. Parameters for Message Transactions (PB2-20) (Cont.)

PH4 MESSAGE TYPE	MESSAGE	PB1 BLUE UNIT ID	PH1 RED UNIT ID	PB3 BLUE ID SENDING MSG.	PB4 BLUE ID RECEIVING MSG.	PB5 AUXILIARY MSG. DATA
103	Self Assign Request	VF	Target	VF	Air Contrl	-
104	Self Assign OK	"	"	Air Contrl	PB1	-
105	Self Assign NoGo	"	"	"	"	-
106	On Self Control	"	"	VF	Air Contrl	-
107	Red Fighter (1)	-	"	"	"	-
	(2)	-	"	Air Contrl	CC	-
108	Intercept Handover	VF	"	CC	Air Contrl	-
109	Accept Handover	"	"	Air Contrl	CC	1=Saturated Now 2=Can accept more
110	VF Launch Order	-	0 if CAP Target if DLI	CC	CC	0 if DLI CAP Station No. if CAP
111	VF Engaging Target	VF	Target	Air Contrl	CC	-
112	Reject Handover	"	"	Air Contrl	CC	-
113	CANTCO	CV	"	CV	CC	TYPE A/C
114[1]	Cancel Launch	CV	0	CC	CV	Type A/C
115[3]	Revise Launch	CV	0	CC	CV	Type A/C
117[2]	Scheduled Launch	CV	Target	CC	CV	Type A/C
202	Assign Nuc	"	"	"	"	-
203	Self Assign	"	"	PB1	CC	-
204	Self Assign Nuc	"	"	"	"	-
205	Self Assign NoGo	"	"	CC	PB1	-
206	Self Assign Notice	"	"	PB1	CC	-

Footnotes:
1 PH-6 Time Launch Scheduled
2 PB-6 CAP Number PH5 - Time Launch Desired
3 PH-6 Time Launch Currently scheduled
PH-5 Time Launch Desired

3.4.2 Events Scheduled by Transactions

For transaction types other than messages, parameter P4 is frequently used to identify the type of event scheduled to occur next for the transaction. (P4 is not used for certain transaction types that are associated with a single event type and in other instances where the event type is not applicable.) The event codes appearing in P4 are defined in Table 3-6.

Table 3-7 identifies all events that are scheduled via transactions on the user chain. The events are organized by the module that schedules them. The transaction types used for each event and the contents of the principal parameters are defined, including the GPSS block label/address where the event will be processed.

Events in the 500 series are unique to aircraft handling on the aircraft carriers.

Table 3-6. Transaction Event Codes (P4)

The following Transaction Event Codes represent the following events of the GPSS Program:

500 - No future event scheduled.

501 - Hook up to carrier launcher.

502 - Increase alert level.

503 - Step down from alert condition.

504 - Aircraft up -- returns to operational readiness status.

505 - Land aircraft on carrier.

506 - Launch aircraft.

511*- Hook up/immediate launch aircraft.

888 - Issued updated launch aircraft.

900 - Track/action decision.

901 - New detection, Red course change, State 5 (ready for self-intercept), sudden loss of detection.

902 - No response to self-assign request.

903 - Weapon launch.

904 - End of dogfight, VF wins.

905 - End of dogfight, VF loses.

906 - VF on CAP (arrives on station or arrives at expected intercept point without detecting target).

907 - VF returns to CV because of low fuel.

908 - Blue Unit loses detection on Red target.

909*- Termination of Blue Unit (killed or no longer able to carry out mission).

910*- No change in previously scheduled event.

911 - Blue conventional missile (AIM or SAM) hits Red target.

912 - Blue missile misses Red target.

913 - End of VF slowdown to let Red target move away for use of long range weapon.

914*- Red Unit that is an active target for this Blue Unit killed by another Blue Unit or left game.

915*- VF is waiting for a response to a self-assign request for this target.

916*- Code returned by VFCALC for Event 914 when VF is already on CAP (awaiting response to self-assign for this Red Unit).

917 - Blue fighter shot down in dogfight.

918 - Nuclear SAM hits Red target.

919 - Aircraft arrive at altitude.

920 - End of SAM firing evaluation (for miss).

921 - End of SAM firing evaluation (for hit).

922*- Invalid current event.

923 - Planned Blue aircraft maneuver.

3.5 FORTRAN ORGANIZATION

3.5.1 Program Hierarchy

A single interface between the GPSS of NADS and the supporting FORTRAN subroutines is furnished. The GPSS program calls HLPRTN whenever FORTRAN compurations are required. One of the arguments passed by GPSS is the identifier

----

* These codes do not represent real events but are used to help route transactions through the GPSS program.

of the primary FORTRAN routine required.

The relationships between the subprograms for the FORTAN can be determined from the map produced by the linker when the programs are linked.

3.5.2 User Data Files

The user data data files are listed below with their associated file numbers. The NADS Users' Manual should be referenced for the format content of each file.

11 - Blue Sensor Characteristics

12 - Blue Aircraft Characteristics

13 - Blue Aircraft Missile Characteristics

14 - Blue Ship Characteristics

15 - Blue SAM Characteristics

16 - Red Aircraft Characteristics

17 - Red Missile Characteristics

18 - Nuclear Warhead Characteristics

19 - Miscellaneous

20 - CV VF Launch Characteristics/Status

21 - Blue Units

22 - CAP Stations

23 - Red Aircraft Scenario

24 - Red SSM Scenario

25 - Jammer Characteristics

32 - Blue Aircraft Characteristics - Nuclear Vulnerability

34 - Blue Ship Characteristics - Nuclear Vulnerability

36 - Red Aircraft Characteristics - Nuclear Vulnerability

39 - Intelligence Data for Red Aircraft

40 - External Surveillance Messages

3.5.3 Programs Versus Common Blocks

The FORTRAN common variables are organized into 35 blocks of labelled common. Each labelled common is contained in a separate computer file and is "included" in the desired subprograms. The pages that follow are arranged with the common block labels in alphabetical order. These pages are listed directly from the associated computer file to facilitate documentation accuracy. Intermal to the file each variable in the block is documented.

Most of the FORTRAN common variables are named so that the first two or three letters of the name indicate the labelled common containing the variable.

The data types (real,integer, logical) and length (in bytes) of each variable is defined. The righthand column identifies the initialization of each variable that requires it. Constant values are initialized in NADSBLOCK.DAT, the block data program. Where the initial value depends on the input data, the input file number is shown. (These files are named in Section 3.5.2). In some instances the initial value is computed in the program INIT, using data from two or more input files, or in some other program than INIT, as indicated.

To facilitate future updates, insertions, and deletions, the table dedicates a new page to each of the labelled blocks.

```
*                                                                        *
* ---------------------------------------------------------------------- *
*    AJPLAN COMMON BLOCK                         08/14/84                *
* ---------------------------------------------------------------------- *
*                         active jamming status                          *

*                                            CONTENT          INITIAL VALUE
*                                                                OR FILE


      PARAMETER  NJ_ACT = BL_IDS + VQ_ID
                                   ! Total number of Blue units
                                   ! (ship or VAQ) jamming


      REAL*4     AJWMH( NJ_ACT, 25 )
                                   !Power density of jammer, W/MHZ      F-30

      INTEGER*2  AJBUID( NJ_ACT),  ! Blue identification of the         F-30
                                   ! jamming platform

     .           AJNUM,            ! Number of active jamming plan
                                   ! Blue unit types

     .           AJVICT( NJ_ACT ), ! number of victims to a blue        F-30
                                   ! unit
     .           AJVCNO( NJ_ACT, 25 )
                                   ! the victim sensor  number          F-30
                                   ! ( of the number of victims)


      COMMON /AJPLAN/ AJNUM, AJVICT, AJVCNO, AJBUID, AJWMH
```

```
*                                                                        *
* ---------------------------------------------------------------------- *
*     APSTAT COMMON BLOCK                        05/21/85                *
* ---------------------------------------------------------------------- *
*                              AirPlane status                           *

*                                                CONTENT      INITIAL VALUE
*                                                                OR FILE


**    PARAMETER  AP_ID = VW_ID + VQ_ID + VF_ID
**                                 ! APID = blue unit ID - APBIAS        **


      REAL*4  APXC ( AP_ID ),      ! x-coordinate                   F-21

     .        APYC ( AP_ID ),      ! y-coordinate                   F-21

     .        APZC ( AP_ID ),      ! z-coordinate                   F-21

     .        APXS ( AP_ID ),      ! x-velocity                     ---

     .        APYS ( AP_ID ),      ! y-velocity                     ---

     .        APZS ( AP_ID ),      ! z-velocity                     ---

     .        APFBR ( AP_ID ),     ! fuel burn rate                 ---

     .        APFREM ( AP_ID )     ! fuel remaining at time of      F-22
                                   ! last update


      INTEGER*2  APTLU ( AP_ID ),  ! time of last position update   F-21

     .           APCID ( AP_ID ),  ! blue unit ID of controller     F-22

     .           APBIAS,           ! highest ID number assigned to  F-21
                                   ! blue units other than AP
                                   ! aircraft

     .           APCNT,            ! number of aircraft which have  F-21/22
                                   ! entered game

     .           APPNTR ( AP_ID ), ! The pointer that links the aircraft
                                   ! id to the type id for use in
                                   ! accessing the VFSTAT, VWSTAT and
                                   ! VQSTAT commons.
```

```
     .          APCVID ( AP_ID )    ! blue unit ID of home carrier        F-22


      LOGICAL*1  APREF ( AP_ID ),     ! refuelled flag ( T = has been       F
                                      ! refuelled)

     .          APTURN( AP_ID)        ! aircraft turn indicator             F

      COMMON /APSTAT/ APXC, APYC, APZC, APXS, APYS, APZS, APFBR, APFREM,
     .                APTLU, APCID, APBIAS, APCNT, APCVID, APPNTR,
     .                APREF, APTURN
```

```
*                                                                        *
* ---------------------------------------------------------------------- *
*     BASTAT COMMON BLOCK                         07/02/84               *
* ---------------------------------------------------------------------- *
*                      blue aircraft status                              *


*                                               CONTENT          INITIAL VALUE
*                                                                   OR FILE


**    PARAMETER  VFC_PT = 10,        ! max number of blue fighter platforms  **
**                                                                           **
**    .          VQC_PT = 5,         ! max number of blue electronic warfare **
**                                   ! platforms                             **
**                                                                           **
**    .          VWC_PT = 5,         ! max number of blue early warning      **
**                                   ! platforms                             **
**                                                                           **
**    .          AC_ID = VFC_PT + VQC_PT + VWC_PT                            **
**                                   ! the possible total number of blue     **
**                                   ! aircraft platform types               **

      INTEGER*2  BAFTY( AC_ID),
                                     ! blue aircraft functional type      F-07
                                     !                                    F-09
                                     !                                    F-12

      .          BANUM,
                                     ! number of blue aircraft            F-07
                                     !                                    F-09
                                     !                                    F-12

      .          BAPTY( AC_ID),
                                     ! blue aircraft platform type        F-07
                                     !                                    F-09
                                     !                                    F-12

      .          BATYP( 2:4, VFC_PT )  ! blue aircraft type pointer         F-07
                                     !                                    F-09
                                     !                                    F-12

      COMMON /BASTAT/ BAFTY, BANUM, BAPTY, BATYP
```

```
*                                                                       *
* --------------------------------------------------------------------- *
*    BFCHAR COMMON BLOCK                            05/24/84            *
* --------------------------------------------------------------------- *
*                   Blue Fire Control Characteristics                   *

*                                              CONTENT       INITIAL VALUE
*                                                               OR FILE

     PARAMETER  BF_CT = 20          ! blue fire control type

     REAL*4     BFCRIL ( BF_CT )    ! tieup range, (nmi)                 F-10


     INTEGER*2  BFCFCS ( BF_CT ),   ! number of fire control             F-10
                                    ! system channels
    .           BFCILL ( BF_CT ),   ! number of SAM illuminators         F-10
                                    !  ( or guidance channels )
    .           BFCTIL ( BF_CT, 2 ), ! illuminator tie-up times for      F-10
                                    ! short range and for long range
                                    ! intercepts
    .           BFCTLK ( BF_CT ),   ! lock-on delay for FC radar after   F-10
                                    ! target designation
    .           BFCTWT ( BF_CT )    ! estimated waiting time for an      F-10
                                    ! illuminator, used to predict
                                    ! shoot-look-shoot opportunity


     CHARACTER*10 BFCSNM ( BF_CT )  ! blue fire control system name      F-10


     COMMON /BFCHAR/ BFCFCS, BFCILL, BFCRIL, BFCSNM, BFCTIL, BFCTLK,
    .                BFCTWT
```

```
*                                                                        *
* --------------------------------------------------------------------- *
*    BJCHAR COMMON BLOCK                         07/11/84                *
* --------------------------------------------------------------------- *
*                        Blue Jammer Characteristics                     *

*                                              CONTENT         INITIAL VALUE
*                                                                 OR FILE


     PARAMETER  BD_JMT = 8           ! blue jammer type


     REAL*4  BJCPOW ( BD_JMT ),      ! total effective radiated          F-28
                                     !  jamming power, watts

    .        BJCMXW ( BD_JMT ),      ! maximum bandwidth, MHz            F-28

    .        BJCMNW ( BD_JMT )       ! minimum bandwidth, MHz            F-28

     INTEGER*2 BJRSN( BD_JMT, 6),    ! BLUE jamming RED Sensor number    F-28

    .          BJREST                ! BLUE jamming Response Time after  F-28
                                     ! receiving signal.

     LOGICAL*1 BJAUTO                ! Switch to control the response of F-28
                                     ! BLUE aircraft to RED tracking.
                                     ! TRUE = aircraft will initiate
                                     ! jamming if possible when tracked.

     COMMON /BJCHAR/ BJRSN, BJCPOW, BJCMXW, BJCMNW, BJREST,
    .                BJAUTO
```

```
*                                                                            *
* -------------------------------------------------------------------------- *
*    BLCHAR COMMON BLOCK                       04/05/85                      *
* -------------------------------------------------------------------------- *
*             surface-to-air missile launcher characteristics                *

*                                               CONTENT          INITIAL VALUE
*                                                                   OR FILE


      PARAMETER  BS_LT = 20             ! Blue sam launch type


      INTEGER*2  BLCLNO ( BS_LT ),      ! sam launch number             F-08

     .           BLCLAS ( BS_LT ),      ! this number points to the     F-08
                                        ! launcher class
                                        !  ( 1 = single rail,
                                        !    2 = dual rail,
                                        !    3 = vertical launching system )

     .           BLCTLD ( BS_LT ),      ! sam launcher reload time      F-08

     .           BLCLTP ( BS_LT ),      ! sam launch type               F-08

     .           BLCSLW ( BS_LT )       ! average launcher slew time    F-08

      CHARACTER*10 BLCLNM ( BS_LT )     ! blue sam launch system name


      COMMON /BSCHAR/ BLCLNO, BLCTLD, BLCLTP, BLCSLW, BLCLNM, BLCLAS
```

```
*                                                                          *
* ------------------------------------------------------------------------ *
*     BLCOM COMMON BLOCK                          10/20/80                 *
* ------------------------------------------------------------------------ *
*                          work area for nuclear scaling                   *

*                                               CONTENT        INITIAL VALUE
*                                                                 OR FILE


      REAL*4  FACT                      ! scale factor              RTRIR


      INTEGER*2  J,                     ! interpolation index for   RTRIR
                                        !  function TERPT

     .           K                      ! interpolation index for   RTRIR
                                        !  function TERPT


      COMMON /BLCOM/ FACT, J, K
```

```
*                                                                         *
* ----------------------------------------------------------------------- *
*     BLUNIT COMMON BLOCK                          08/30/85               *
* ----------------------------------------------------------------------- *
*                               blue unit status                          *

*                                             CONTENT          INITIAL VALUE
*                                                                 OR FILE


**    PARAMETER  BL_IDS = 60,          ! blue unit ID                      **
**                                     !  ( any ship unit )                **
**                                                                         **
**    .          BL_ID  = AP_ID + BLIDS
**                                     ! blue unit ID                      **
**                                     !  ( any type of unit )             **
**                                                                         **
**    .          RD_ID  = 510,         ! red unit ID                       **
**                                                                         **
**    .          NJ_BLU = 6,           ! number of jammers per blue unit   **
**                                                                         **
**    .          BL_IDC = BL_IDS + VW_ID                                   **
**                                     ! the total possible number of blue **
**                                     ! aircraft control units            **


      REAL*4  BUXC ( BL_IDS ),         ! x-coordinate                 F-21

      .       BUYC ( BL_IDS ),         ! y-coordinate                 F-21

      .       BUZC ( BL_IDS ),         ! z-coordinate                 F-21

      .       BUTNKR                   ! tanker fuel available        F-19


      INTEGER*2  BUFTY ( BL_ID ),      ! functional type            0/F-21
                                       !  ( 1=ship,
                                       !    2=VAW,
                                       !    3=VAQ,
                                       !    4=VF,
                                       !    negative=killed )

      .          BUPTY ( BL_ID ),      ! platform type                F-21
                                       !  ( index to
                                       !  characteristics arrays )

      .          BUPSEN ( BL_ID ),     ! air search radar sensor      F-21
                                       !  type
```

```
        BUNSEN ( BL_ID ),     ! number of sensors in up         INIT
                              !  status

        BUNUCT ( BL_ID ),     ! time that maximum damage          0
                              !  occurred

        BUDMG ( BL_ID ),      ! level of nuclear maximum damage:  0
                              !   ships  0 to 6,
                              !   A/C  to 2

        BUNUCJ ( BL_ID ),     ! nuclear environment that          0
                              !  caused maximum damage:
                              !    0 = nonnuc
                              !    1 to NC_ENV for nuclear

        BUTDNV ( BL_ID ),     ! blue unit time detect not     32767
                              ! valid

        ACCTVF ( BL_IDC ),    ! count of VF assigned to           0
                              !  air controller  BL_IDC

        ACCTVC ( BL_IDC ),    ! count of VF being vectored        0
                              !  by controller  BL_IDC

        ACMXVF ( BL_IDC ),    ! max number of VF that can be   F-21
                              !  assigned to  BL_IDC

        ACMXVC ( BL_IDC )     ! max number of VF that          F-21
                              !   controller  BL_IDC  can vector


LOGICAL*1  BUTRK ( BL_ID , RD_ID ),
                              ! track indicator ( true if     FALSE
                              !  blue unit BL_ID is tracking
                              !  red unit RD_ID)

        BUSSJ ( BL_ID , RD_ID ),
                              ! self screening indicator( true FALSE
                              !  if BLUE unit is being jammed
                              !  by RED unit)

        BUTDS ( BL_ID )       ! tactical data system       F-12,14
                              !  capability
                              !  ( T = up, F = down )
```

```
COMMON /BLUNIT/ BUXC, BUYC, BUZC, BUFTY, BUPTY, BUTNKR, BUPSEN,
.               BUNSEN, BUDLEVL, BUNUCT, BUDMG, BUNUCJ, BUTDNV,
.               ACCTVF, ACCTVC, ACMXVF, ACMXVC,
.               BUTRK, BUSSJ, BUTDS
```

```
*                                                                        *
* ---------------------------------------------------------------------- *
*    BMCHAR COMMON BLOCK                        10/18/85                 *
* ---------------------------------------------------------------------- *
*              blue VF missile characteristics and LAR data              *

*                                            CONTENT          INITIAL VALUE
*                                                               OR FILE


      PARAMETER  TN_MSL = 6,          ! type number of missile

    .            MC_MAX = 10          ! maximum number of missiles that
                                      ! any fighter can carry


     REAL*4  BMCRMX ( TN_MSL ),       ! maximum range                 F-13

    .        BMCRMN ( TN_MSL ),       ! minimum range                 F-13

    .        BMCAMX ( TN_MSL ),       ! maximum altitude              F-13

    .        BMCVEL ( TN_MSL ),       ! average horizontal velocity   F-13

    .        BMCPK ( TN_MSL, 2 ),     ! probability of kill           F-13
                                      ! one is the pk in active or semi-
                                      ! active homing, two is the pk in
                                      ! home-on-jam mode

    .        BMHOJF ( TN_MSL ),       ! LAR reduction factor for      F-13
                                      !  home-on-jam mode

    .        BMLTS ( TN_MSL, 3 ),     ! three target speeds, for      F-13
                                      !  LAR computations

    .        BMLDZ ( TN_MSL, 3 ),     ! delta Z    ( 3 altitude       F-13
                                      !  differences for LARs )

    .        BMLAOB ( TN_MSL, 3, 3, 6 ),
                                      ! angle on bow, 6 values for    F-13
                                      !  each combination of target
                                      !  speed and delta Z

    .        BMLRLA ( TN_MSL, 3, 3, 6 )
```

```
                                  ! radius of LAR, 6 values for       F-13
                                  !  each combination of target
                                  !  speed and delta Z


    INTEGER*4  BMCNAM ( TN_MSL, 2 )
                                  ! eight-character name for          TBD
                                  !  the missile type


    INTEGER*2  BMCWH ( TN_MSL )   ! warhead type ( 0 = nonnuc,        F-13
                                  !  >0 = index to nuclear warhead
                                  !  characteristics arrays )


    COMMON /BMCHAR/ BMCRMX, BMCRMN, BMCAMX, BMCVEL, BMCPK, BMHOJF,
   *                BMLTS, BMLDZ, BMLAOB, BMLRLA, BMCNAM, BMCWH
```

```
*                                                                     *
* ------------------------------------------------------------------- *
*     CCGRP COMMON BLOCK                      11/01/85                *
* ------------------------------------------------------------------- *
*              command center perceived status of red groups          *


*                                          CONTENT          INITIAL VALUE
*                                                            OR FILE


*    PARAMETER  RD_IDM =  30       ! red group ( or track ) ID number


     REAL*4  CGXC ( RD_IDM ),       ! x-coordinate of the most          SURMSG
                                    !  recently observed position

     .       CGYC ( RD_IDM ),       ! y-coordinate of the most          SURMSG
                                    !  recently observed position

     .       CGXRKO ( RD_IDM ),     ! x-coordinate of the CAP station   SURMSG
                                    !  that is positioned in response
                                    !  to data on red group

     .       CGYRKO ( RD_IDM )      ! y-coordinate of CAP station       SURMSG
                                    !  positioned for red group


     INTEGER*4  CGGID ( RD_IDM )    ! red group identifier              SURMSG


     INTEGER*2  CGTOBS ( RD_IDM ),  ! time of most recent observation    0
                                    !  of red group

     .          CGTAS ( RD_IDM ),   ! earliest time that red group is   SURMSG
                                    !  expected to arrive at the
                                    !  associated CAP station

     .          CGCNT ( RD_IDM ),   ! count of aircraft in the group,    0
                                    !  reported or assumed

     .          CGGEN ( RD_IDM ),   ! generic airframe label for group   0
                                    !  members; index to intelligence
                                    !  arrays ( reported or assumed )
```

```
.            CGFT ( RD_IDM ),      ! functional type of red group       0
                                   !  members  ( 1 = bomber,
                                   !             2 = fighter,
                                   !             3 = recce,
                                   !             4 = SOJ )

.            CGTCNT ( RD_IDM ),    ! time at which count was last       0
                                   ! observed

.            CGTGEN ( RD_IDM ),    ! time at which airframe label       0
                                   ! was last observed

.            CGTFT ( RD_IDM ),     ! time at which functional type      0
                                   ! was last observed

.            CGVAW ( RD_IDM ),     ! station identification ( IC)       0
                                   ! for the early warning station
                                   ! which will cover this group

.            CGTOT                 ! total number of red groups         0
                                   !  currently observed


   LOGICAL*1 CGLCOV ( RD_IDM ),    ! T = group adequately covered     FALSE
                                   ! F = inadequate

.            CGLNOC ( RD_IDM )     ! T = no additional coverage       FALSE
                                   !        required
                                   ! F = more coverage required


   COMMON /CCGRP/ CGXC, CGYC, CGXRKO, CGYRKO, CGGID,
.                 CGTOBS, CGTAS, CGCNT, CGGEN, CGFT,
.                 CGTCNT, CGTGEN, CGTFT, CGTOT, CGVAW,
.                 CGLCOV, CGLNOC
```

```
*                                                                          *
* ------------------------------------------------------------------------ *
*     CCSTAT COMMON BLOCK                       08/31/84                   *
* ------------------------------------------------------------------------ *
*                  command center perceived status of blue units           *


*                                              CONTENT           INITIAL VALUE
*                                                                   OR FILE


**    PARAMETER  BL_IDS =   60,       ! blue unit ID                       **
**                                    !  ( for any ship unit )             **
**                                                                         **
**    .          VF_ID  =   90,       ! VF ID number                       **
**                                                                         **
**    .          BL_ID  =  AP_ID + BL_IDS
**                                    ! ID of any blue unit                **
**                                    !  ( BL_ID = VF_ID + VFBIAS )        **
**                                                                         **
**    .          BL_IDC = BL_IDS + VW_ID                                   **
**                                    ! the total possible number of blue  **
**                                    ! aircraft control units             **


      INTEGER*4  CCBUID               ! blue unit containing the      F-19
                                      !  command center


      INTEGER*2  CSNFCC ( BL_IDS ),   ! number of SAM fire control    F-14,21
                                      !  channels free

      .          CSACAV ( BL_IDC ),   ! air controller availability   0/F-21
                                      !  ( 0 = no capability,
                                      !    1 = have capability but full,
                                      !    2 = can control more )

      .          CSACAS ( BL_IDC ),   ! number of VF assigned to      0/NEWAC
                                      !  controller BL_IDC

      .          CSVFAV ( VF_ID ),    ! VF availability               0/APINIT
                                      !  ( 0 = no targets assigned,
                                      !    1 = 1 target assigned,
                                      !    2 = 2 targets assigned,
                                      !    3 = out of game )
```

```
     .           CSACID ( AP_ID ),     ! blue unit ID of controller       0/NEWAC
                                       !  assigned to this VF

     .           CSWUTM                ! game time that next command        32767
                                       !  center decision will be required


      LOGICAL*1  CSCACC ( AP_ID ),     ! assigned controller has accepted,  FALSE
                                       !  true or false

     .           CSUNIT ( BL_ID )      ! unit status                      F/F-21
                                       !  ( T = up,  F = down )


      COMMON /CCSTAT/ CCBUID, CSNFCC, CSACAV, CSACAS, CSVFAV,
     .                CSACID, CSWUTM, CSCACC, CSUNIT
```

```
*                                                                          *
* ------------------------------------------------------------------------ *
*     CCTARG COMMON BLOCK                      08/30/85                    *
* ------------------------------------------------------------------------ *
*                 command center perception of red target status           *

*                                             CONTENT           INITIAL VALUE
*                                                                  OR FILE


**     PARAMETER  RD_ID =  510        ! red unit ID                         **
**     PARAMETER  VF_ID =   90        ! blue fighter number                 **
**     PARAMETER  BL_ID =  AP_ID + BL_IDS
**                                    ! blue unit ID                        **
**     PARAMETER  RD_BOF=  160        ! blue unit ID                        **

       PARAMETER  BL_BOF = RD_BOF * BL_ID
       REAL*4     CTJBER( RD_BOF)     ! jammer bearing between           0.0
                                      ! ( 0, twopi]

       INTEGER*2  CTCATG ( RD_ID ),   ! target category                  0
                                      !  ( -1 = killed,
                                      !     0 = not yet reported,
                                      !     1 = bomber before launch,
                                      !     2 = bomber after launch,
                                      !     3 = SSM,
                                      !     4 = ASM,
                                      !     5 = recce A/C,
                                      !     6 = ECM A/C,
                                      !     7 = fighter )

     .            CTSASN ( RD_ID ),   ! blue ID of ship assigned to      0
                                      !  target by CC ( set to -2 if
                                      !  target is killed or out of game )

     .            CTNTDS ( RD_ID ),   ! number of NTDS participating     0
                                      !  units that are currently
                                      !  reporting contact

     .            CTJCOV ( RD_BOF),   !                                  0
                                      ! Blue unit ID of fighter assigned to
                                      !  cover this active jammer.

     .            CTVFID ( RD_ID )    ! blue unit ID of fighter assigned to  0
                                      !  intercept ( set to -1 for a
                                      !  new launch, set to -2 if target
                                      !  is killed or out of game )
```

```
        LOGICAL*1

       .            CTNLAR ( RD_ID, VF_ID),!                         FALSE
                                       ! LAR not achievable against this red

       .            CTACTJ ( RD_BOF, BL_ID)!                         FALSE
                                       ! Red unit is actively jamming blue unit


        COMMON /CCTARG/ CTJBER, CTCATG, CTSASN, CTNTDS, CTJCOV, CTVFID,
       1                CTNLAR, CTACTJ
```

```
*                                                                              *
* ---------------------------------------------------------------------------- *
*     CONST COMMON BLOCK                           12/18/84                    *
* ---------------------------------------------------------------------------- *
*                                  constants                                   *

*                                                  CONTENT          INITIAL VALUE
*                                                                      OR FILE


*     REAL*4  CONVDG,                  ! degrees to radians conversion   .0174533
                                       !  factor

*    .        CONVFT,                  ! feet to meters conversion factor  .3048

*    .        CONVKF,                  ! kilofeet to meters conversion     304.8
                                       !  factor

*    .        CONVKN,                  ! knots to meters per second       .51444
                                       !  conversion factor

*    .        CONVLB,                  ! pounds mass to kilograms          .4536
                                       !  conversion factor

*    .        CONVLH,                  ! pounds per hour to kilograms    .000126
                                       !  per second conversion factor

*    .        CONVNM,                  ! nautical miles to meters          1852.
                                       !  conversion factor

*    .        RNEVER,                  ! time later than end of game      32767.

*    .        TWOPI,                   ! Pi * 2                        6.2831853

*    .        HLFPI,                   ! Pi * 1/2                      1.5707963

*    .        PI                       ! Pi                            3.1415927


*     INTEGER*4  NEVER                 ! largest half-word value for      32767
                                       !  setting half-word integer
                                       !  to time later than end of game


*     INTEGER*2  CONVMN                ! minutes to seconds conversion     60
                                       !  factor
```

```
*     COMMON /CONST/ CONVDG, CONVFT, CONVKF, CONVKN, CONVLB,
*    .               CONVLH, CONVNM, RNEVER, TWOPI, HLFPI,
*    .               PI, NEVER, CONVMN

C ********************************************************************
C *  CONS COMMON BLOCK INITIALIZATION
C ********************************************************************
C
C****    CONVERSION CONSTANTS    ******
C
C      CONVDG - DEGREES TO RADIANS
C      CONVFT - FEET TO METERS
C      CONVKF - KILOFEET TO METERS
C      CONVKN - KNOTS TO METERS/SECOND
C      CONVLB - POUNDS TO KILOGRAMS
C      CONVLH - POUNDS/HOUR TO KILOGRAMS/SECOND
C      CONVNM - NAUTICAL MILES TO METERS
C      NEVER - LARGEST HALF-WORD VALUE FOR SETTING HALF-WORD TIME
C      CONVMN - MINUTES TO SECONDS (60)
C      CNVTMN - MINUTES TO SECONDS (60.)
C
       PARAMETER PI = 3.1415927, CONVFT = 0.3048, CONVKF = 304.8,
     .   CONVKN = 0.51444, CONVLB = 0.4536, CONVLH = 0.000126,
     .   CONVNM = 1852.0, CONVDG = PI / 180.0, CONVMN = 60,
     .   CNVTMN = 60.0, NEVER = 32767, RNEVER = 32767.0,
     .   TWOPI = PI * 2.0, HLFPI = PI / 2.0
```

```
*                                                                      *
* -------------------------------------------------------------------- *
*    CPSTAT COMMON BLOCK                        08/29/85               *
* -------------------------------------------------------------------- *
*                               blue aircraft stations                 *

*                                             CONTENT       INITIAL VALUE
*                                                              OR FILE


**    PARAMETER  STA_ID = 120        ! station identifier, for fixed
**                                   !  defense posture, or for response
**                                   !  to tactical situation


      REAL*4  CPBEAR( STA_ID),       ! true bearing to station from     0/F-22
                                     ! force center, ZZ

     .        CPRNG( STA_ID),        ! range to station from            0/F-22
                                     ! force center, ZZ

     .        CPSIZE( STA_ID),       ! half-width of the station in     0/F-22
                                     ! radians for back-and-forth
                                     ! patrolling

     .        CPXC ( STA_ID ),       ! X-coordinate of station          0/F-22

     .        CPYC ( STA_ID ),       ! Y-coordinate of station          0/F-22

     .        CPH ( STA_ID )         ! altitude of station              0/F-22


      INTEGER*2  CPGP ( STA_ID ),    ! index of target group in         0/F-22
                                     !  response to which the station
                                     !  is to be filled ( negative is
                                     !  for stations of the fixed
                                     !  defense posture )

     .           CPTIME ( STA_ID ),  ! time for which aircraft has      0/F-22
                                     !  been at this station at start
                                     !  of game

     .           CPBLID ( STA_ID ),  ! blue unit ID of aircraft         0/F-22
                                     !  assigned to the station
```

```
.          CPAFTY ( STA_ID ),  ! functional type of aircraft       O/F-22
                               !  required by the station

.          CPAS ( STA_ID ),    ! alert status of aircraft to       O/F-22
                               !  fill the station

.          CPTOR ( STA_ID ),   ! time at which aircraft should     O/F-22
                               !  respond to fill the station

.          CPTAS ( STA_ID )    ! time at which aircraft should     O/F-22
                               !  arrive at station


      COMMON /CPSTAT/ CPBEAR, CPRNG, CPSIZE, CPXC, CPYC, CPH, CPGP,
     .                CPTIME, CPBLID, CPAFTY, CPAS, CPTOR, CPTAS
```

```
*                                                                          *
* ------------------------------------------------------------------------ *
*    CVSTAT COMMON BLOCK                        09/27/85                   *
* ------------------------------------------------------------------------ *
*                              CC perception of CV assets                  *

*                                      CONTENT                INITIAL VALUE
*                                                                OR FILE

      PARAMETER  CV_IND =  5,        ! CV index

     .           AL_LVL =  3,        ! alert level

     .           BAC_CV = AC_ID * AL_PSG + AC_ID,
                                     ! blue aircraft coverage

     .           CV_PT  = CV_IND * AC_ID,

     .           CV_PTA = CV_PT * AL_LVL

**   PARAMETER   AC_ID = VFC_PT + VQC_PT + VWC_PT                          **
**                                   ! the possible total number of blue   **
**                                   ! aircraft platform types             **
**                                                                         **
**   .           STA_DF = 15,        ! station defined for fixed defense   **
**                                   !  posture  ( I = RAC_LB + 1 to       **
**                                   !  AL_PSG - 1)                        **
**                                                                         **
**   .           AL_PSG = RAC_LB + STA_DF + 1,                             **
**                                   ! total pseudo groups for coverage,   **
**                                   ! red groups and fixed stations for CAP **
**                                   ! AL_PSG is used for early warning    **
**                                   ! responsive stationing               **
**                                                                         **
**   .           RAC_LB = 30         ! red aircraft airframe label         **
**                                   !  ( I = 1 to 30 )                    **


      REAL*4  CVNBCV ( AC_ID, AL_PSG + 1)
                                     ! number of target aircraft      0/F-22,39
                                     !  that aircraft number AC_ID
                                     !  can cover in an intercept
                                     !  mission; set to 1 for aircraft
                                     !  of the types appropriate for
                                     !  each of the fixed stations
```

```
INTEGER*2  CVBUID ( CV_IND ),  ! blue unit ID for the CV           O/F-20

.          CVNAS ( CV_IND, AL_LVL, AC_ID ),
                               ! the ID number of the aircraft     O/F-22
                               !  AC_ID in alert level  AL_LVL
                               !  available on CV CV_IND

.          CVASTM ( CV_IND, AL_LVL, AC_ID ),
                               ! time associated with alert        O/F-22
                               !  level AL_LVL for the number
                               !  of aircraft AC_ID on CV CV_IND

.          CVTOR ( CV_IND, AC_ID ),
                               ! time to schedule launch for     O/CVLNCH
                               !  the number of aircraft AC_ID
                               !  on CV CV_IND, to meet a tactical
                               !  requirement

.          CVTRDX ( CV_IND * AC_ID ),
                               ! list of packed indices for the  O/CVLNCH
                               !  CV alert level, ordered from
                               !  largest to smallest time to
                               !  schedule launch

.          CVCNT,              ! number of CVs in the game         O/F-20

.          CVACT,              ! number of CVs still active         F-20

.          CVTORN              ! number of CV-alert level pairs  O/CVLNCH
                               !  in the list of packed indices

LOGICAL*1

.          CVUP( CV_IND),      ! indicator that the carrier is
                               ! still active

.          CVWATA( CV_IND, AC_ID)
                               ! indicator that the number of .FALSE./F-22
                               !  aircraft is assigned to a carrier
                               !  air wing


COMMON /CVSTAT/ CVNBCV, CVBUID, CVNAS, CVASTM, CVTOR,
.               CVTRDX, CVCNT, CVACT, CVTORN, CVUP, CVWATA
```

```
*                                                                       *
* --------------------------------------------------------------------- *
*     DTCT COMMON BLOCK                            08/04/86             *
* --------------------------------------------------------------------- *
*                          DETECT computations                          *

*                                              CONTENT        INITIAL VALUE
*                                                               OR FILE


      REAL*8  DTACON,             !  coefficients of a polynomial

     .        DTBCON,             !  coefficients of a polynomial

     .        DTCCON,             !  coefficients of a polynomial

     .        DTDCON,             !  coefficients of a polynomial

     .        DTECON,             !  coefficients of a polynomial

     .        DTAVAR( RD_BOF),    !  coefficients of a cubic

     .        DTBVAR( RD_BOF),    !  coefficients of a cubic

     .        DTCVAR( RD_BOF)     !  coefficients of a cubic

      REAL*4  DTFACT              !  Product of RADAR cross section and
                                  !  Range Alpha

      INTEGER*2  DTNJAM,          !  number of jammers

     .           DTRISE( RD_ID, BL_ID),
                                  !  time RED unit rises above the BLUE
                                  !  unit radio horizon

     .           DTSET( RD_ID, BL_ID),
                                  !  time RED unit sets below the BLUE
                                  !  unit radio horizon

     .           DT1CRS( RD_ID, VF_ID),
                                  !  time the RED unit first crosses the
                                  !  edge of the BLUE aircraft sensor.

     .           DT2CRS( RD_ID, VF_ID)
                                  !  time the RED unit last crosses the
                                  !  edge of the BLUE aircraft sensor.
```

```
COMMON /DTCT/ DTACON, DTBCON, DTCCON, DTDCON, DTECON,
.             DTAVAR, DTBVAR,  DTCVAR, DTFACT, DTNJAM,
.             DTRISE, DTSET, DT1CRS, DT2CRS
```

```
*                                                                      *
* -------------------------------------------------------------------- *
*     ENV COMMON BLOCK                        09/17/86                 *
* -------------------------------------------------------------------- *
*                          nuclear environments                        *

*                                          CONTENT            INITIAL VALUE
*                                                                OR FILE


     REAL*4

   .        DX,                     ! Vector of the relative x-position
                                    ! of the target to the blast

   .        DY,                     ! Vector of the relative y-position
                                    ! of the target to the blast

   .        DZ,                     ! Vector of the relative z-position
                                    ! of the target to the blast

   .        ENV ( NC_ENV ),         ! Intensity of environment
                                    ! at the position of interest

   .        RO,                     ! Initial radius of the fireball

   .        SPEED,                  ! Speed of victim

   .        TENV ( NC_ENV ),        ! Time that environment
                                    ! ENV (NC_ENV ) occurred

   .        TS,                     ! Time the shock arrives

   .        VX,                     ! X-velocity of victim

   .        VY,                     ! Y-velocity of victim

   .        VZ,                     ! Z-velocity of victim

   .        ZP                      ! Z-position of victim at
                                    ! time of burst

     COMMON /ENV/ DX, DY, DZ, ENV, RO, SPEED, TENV, TS,
   .              VX, VY, VZ, ZP
```

```
*                                                                          *
* ------------------------------------------------------------------------ *
*    EXTMSG COMMON BLOCK                        10/17/85                   *
* ------------------------------------------------------------------------ *
*                                                                          *

*                                               CONTENT          INITIAL VALUE
*                                                                   OR FILE


      REAL*4  EXXC,                  ! x-coordinate of observed          F-40
                                     !  position of target group

    .         EXYC,                  ! y-coordinate of observed          F-40
                                     !  position of target group

    .         EXZC,                  ! observed altitude of target group F-40

    .         EXRXU,                 ! standard deviation of observed    F-40
                                     !  position along semi-major axis
                                     !  of uncertainty ellipse

    .         EXRYU,                 ! standard deviation of observed    F-40
                                     !  position along semi-minor axis
                                     !  of uncertainty ellipse

    .         EXRHOU                 ! orientation of semi-major axis    F-40
                                     !  of uncertainty ellipse,
                                     !  relative to grid north


      INTEGER*4  EXGID               ! target group identifier           F-40


      INTEGER*2  EXTRCV,             ! time that command center is to    32767
                                     !  receive message

    .            EXTOBS,             ! time that target group was        F-40
                                     !  observed

    .            EXDT,               ! delta time to adjust message time F-40
                                     !  to correspond to DTIME in File 23
```

```
     .           EXCNT,               ! number of aircraft observed in      F-40
                                      !  target group ( 0 = unknown )

     .           EXGEN,               ! airframe label ( generic type )     F-40
                                      !  observed in the target group
                                      !  ( 0 = unknown )

     .           EXFT                 ! functional type identified in       F-40
                                      !  the target group
                                      !   ( 0 = unknown,
                                      !     1 = bomber,
                                      !     2 = fighter,
                                      !     3 = recce,
                                      !     4 = SOJ )


      COMMON /EXTMSG/ EXXC, EXYC, EXZC, EXRXU, EXRYU,
     .                EXRHOU, EXGID, EXTRCV, EXTOBS, EXCNT,
     .                EXDT, EXGEN, EXFT
```

```
*                                                                         *
* ----------------------------------------------------------------------- *
*     ICWORK COMMON BLOCK                     10/17/85                    *
* ----------------------------------------------------------------------- *
*                                                                         *

*                                             CONTENT          INITIAL VALUE
*                                                                 OR FILE


**    PARAMETER  STA_ID                 ! station identifier, for fixed
**                                      ! defensive posture, or for response
**                                      ! to tactical situation
**
**   .           AL_PSG = RAC_LB + STA_DF + 1,
**                                      ! total pseudo groups for coverage,
**                                      ! red groups and fixed stations for CAP
**                                      ! AL_PSG is used for early warning
**                                      ! responsive stationing

      PARAMETER  SP_CAN = 16,           ! saved preempt candidate or closest
                                        !  non-feasible solution

     .           TGT_GP = STA_ID + SP_CAN
                                        ! target group or station from which
                                        !  VF has been preempted


      REAL*4  ICXS,                     ! x-coordinate of station to be
                                        !  filled

     .        ICYS,                     ! y-coordinate of station to be
                                        !  filled

     .        ICRNG,                    ! range of the station from force
                                        !  center

     .        ICNCPR,                   ! number of target aircraft covered
                                        !  by the saved preempt candidates

     .        ICBRNF,                   ! bearing of the station for VAW
                                        !  for the saved closest non-feasible
                                        !  solution

     .        ICNCNF                    ! number of target aircraft covered
                                        !  by the saved closest non-feasible
                                        !  solutions
```

```
      INTEGER*2  ICTRNF( SP_CAN),      ! time that VF should respond to
                                       !  fill the station, for saved
                                       !  closest non-feasible solution
     .           ICNTNF( SP_CAN),      ! type of VF to fill the station,
                                       !  for saved closest non-feasible
                                       !  solution
     .           ICBUNF( SP_CAN),      ! CV number, or blue unit ID of
                                       !  airborne VF, for saved closest
                                       !  non-feasible solution
     .           ICTANF( SP_CAN),      ! Time of arrival at the station
                                       ! for saved closest
                                       ! non-feasible solution
     .           ICASNF( SP_CAN),      ! Alert state for aircraft on
                                       ! CV for saved closest
                                       ! non-feasible solution
     .           ICCODE( STA_ID),      ! message type to be issued
     .           ICAUX( STA_ID),       ! auxilliary message data
     .           ICPRIG( AL_PSG),      ! index of a target group( or of
                                       !  a fixed station ) from which a
                                       !  VF has been preempted
     .           ICNTPR( SP_CAN),      ! type of VF to fill the station,
                                       !  for the saved preempt candidate
     .           ICICPR( SP_CAN),      ! index of the station position for
                                       !  the saved preempt candidate
     .           ICTRPR( SP_CAN),      ! time that the VF should respond to
                                       !  fill the station, for the saved
                                       !  preempt candidate
     .           ICPRM,                ! number of target groups and fixed
                                       !  stations from which VFs have been
                                       !  preempted
     .           ICNPRM,               ! number of saved preempt candidates
     .           ICNNF,                ! number of saved non-feasible
                                       !  solutions
```

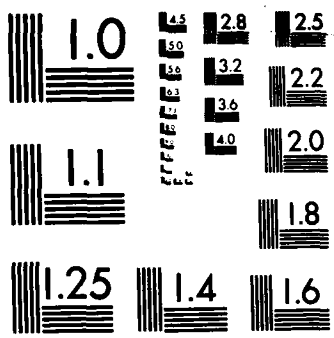
```
     .            ICIG,               ! index for target group status arrays,
                                      !  if positive;  -index for fixed
                                      !  defense posture station, if negative

     .            ICGEN,              ! 1 to 20 = red aircraft airframe label;
                                      ! 21 to 35 = 20+ fixed station index

     .            ICTAS,              ! time that VF should fill the station

     .            ICTNOW              ! current clock time

      EQUIVALENCE (ICNCPR, ICBRNF)


      COMMON/ICWORK/ ICXS, ICYS, ICRNG, ICBRNF, ICNCNF, ICTRNF,
     .       ICNTNF, ICBUNF, ICCODE, ICAUX, ICPRIG, ICNTPR,
     .       ICICPR, ICTRPR, ICPRM, ICNPRM, ICNNF, ICIG, ICGEN,
     .       ICTAS, ICTNOW, ICTANF, ICASNF
```

```
*                                                                          *
* ------------------------------------------------------------------------ *
*    INTEL COMMON BLOCK                         07/31/85                   *
* ------------------------------------------------------------------------ *
*                            intelligence on red targets                   *

*                                               CONTENT          INITIAL VALUE
*                                                                   OR FILE

      PARAMETER INT_MXG = 20           ! maximum number of generic types
                                       ! for the intelligence file.

      REAL*4  INTPSE ( INT_MXG ),      ! estimated penetration speed        F-39

     .        INTRKO ( INT_MXG ),      ! keep-out range                     F-39

     .        INTDST ( 4, INT_MXG ),   ! distance table for worst           F-39
                                       !  case assessments in absence
                                       !  of classification data
                                       !  ( 1 = bomber,
                                       !    2 = fighter,
                                       !    3 = recce,
                                       !    4 = SOJ )

     .        INTMLR                   ! estimated maximum launch range    F-39
                                       !  for red bombers


      INTEGER*2  INTNE ( INT_MXG ),    ! number expected in a red group     F-39
                                       !  if no raid count

     .           INTFTS ( 4, INT_MXG ),
                                       ! up to 4 red functional types      F-39
                                       !  applicable to the indicated
                                       !  airframe label

     .           INTNDX ( 4, INT_MXG ),
                                       ! by functional type, the airframe  F-39
                                       !  label to be assumed for observed
                                       !  distances that equal or exceed
                                       !  INTDST ( 4, INT_MXG )

     .           INTGEN                ! total number of red airframe     0/INIT
                                       !  labels


      COMMON /INTEL/ INTPSE, INTRKO, INTDST, INTMLR, INTNE,
     .               INTFTS, INTNDX, INTGEN
```

```
*                                                                          *
* ------------------------------------------------------------------------ *
*     MESSG COMMON BLOCK                              01/02/86             *
* ------------------------------------------------------------------------ *
*                                    messages                              *


*                                       CONTENT                INITIAL VALUE
*                                                                 OR FILE


      INTEGER*2  MSG(15,9),     ! message parameters for up to 15
                                !  messages to be sent; used by
                                !  FORTRAN subroutines; HLPRTN
                                !  transfers parameters to GPSS MSG
                                !  array prior to returning to GPSS
                                !   1 -- message type (PH4)
                                !   2 -- subject of message (PH7/PB1)
                                !   3 -- blue addressee (PH10/PB4)
                                !   4 -- subject red ID (PH1)
                                !   5 -- auxilliary message data (PH11/PB5)
                                !   6 -- CAP station number (PJ12/PB6)
                                !   7 -- time to launch aircraft (PH5)
                                !   8 -- original action time (PH6)
                                !   9 -- time of message delivery (PH2)

     .           NOMSG          ! number of messages in MSG array;      0
                                !  set to zero on each call to
                                !  HLPRTN


      COMMON /MESSG/ MSG, NOMSG
```

```
*                                                                        *
* ---------------------------------------------------------------------- *
*    MVE COMMON BLOCK                              08/29/85              *
* ---------------------------------------------------------------------- *
*                               move list                                *


*                                            CONTENT         INITIAL VALUE
*                                                               OR FILE


**    PARAMETER  AP_ID =  67,        ! number of aircraft                **
**                                                                       **
**   .           RD_ID = 510         ! number of red units               **


      INTEGER*2  BLUID ( AP_ID ),    ! list of VF IDs for which position
                                     !  and fuel update is to be performed

     .           REDID ( RD_ID )     ! list of red IDs for which position
                                     !  update is to be performed


      COMMON /MVE/ BLUID, REDID
```

AD-A174 818 DESIGN NOTEBOOK FOR NAVAL AIR DEFENSE SIMULATION (NADS) 2/3
REVISION(U) TRW DEFENSE SYSTEMS GROUP MCLEAN VA
WATERWHEEL PROGRAM OFFICE 09 OCT 86 N00014-83-C-0527
UNCLASSIFIED F/G 15/3 NL


1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

```
*                                                                        *
* ---------------------------------------------------------------------- *
*    NJCHAR COMMON BLOCK                         05/01/84                *
* ---------------------------------------------------------------------- *
*                          active jamming status                         *

*                                                CONTENT    INITIAL VALUE
*                                                              OR FILE


     PARAMETER  NJ_ACT = 127


     INTEGER*2  NOVICT( NJ_ACT ),   ! number of victims to a blue    F-30
                                    ! unit
    .           NVICNO( NJ_ACT, 80 )
                                    ! the victims number ( of the    F-30
                                    ! number of victims)


     COMMON /NJCHAR/ NOVICT, NVICNO
```

```
*                                                                        *
* ---------------------------------------------------------------------- *
*     NUCLOG COMMON BLOCK                          03/04/81              *
* ---------------------------------------------------------------------- *
*                        log of nuclear bursts                           *

*                                               CONTENT            INITIAL VALUE
*                                                                    OR FILE


      PARAMETER  IND_BR =  150        ! index number of burst


      REAL*4  NUCTIM ( IND_BR ),      ! time of nuclear burst

     .        NUCX ( IND_BR ),        ! x-position of burst,
                                      !  kilometers

     .        NUCY ( IND_BR ),        ! y-position of burst

     .        NUCZ ( IND_BR ),        ! z-position of burst

     .        NUCTFB ( IND_BR )       ! time that fireball masking
                                      !  effect fades out


      INTEGER*2  NUCTYP ( IND_BR ),   ! warhead type number

     .           NUCN,                ! index of latest entry               0
                                      !  ( number of bursts )

     .           NUCI                 ! index of earliest burst whose       1
                                      !  fireball could still  be active


      COMMON /NUCLOG/ NUCTIM, NUCX, NUCY, NUCZ, NUCTFB,
     .                NUCTYP, NUCN, NUCI
```

```
*                                                                       *
* --------------------------------------------------------------------- *
*    NWCHAR COMMON BLOCK                         04/12/84               *
* --------------------------------------------------------------------- *
*                  nuclear warhead characteristics                      *


*                                          CONTENT           INITIAL VALUE
*                                                               OR FILE


      PARAMETER  WH_TNO =  10        ! warhead type number


      REAL*4  EFTHYLD( WH_TNO),      ! effective thermal yield           F-18
     .        GAMDT ( WH_TNO ),      ! effective prompt gamma pulse      F-18
                                     !  width, seconds
     .        GDOSE,                 ! gamma fluence-to-dose             F-18
                                     !  conversion factor
     .        NWCYLD ( WH_TNO ),     ! burst yield, kilotons             F-18
     .        SG ( WH_TNO ),         ! effective gamma source level      NUCSLEV
     .        SN ( WH_TNO ),         ! neutron source level, n           NUCSLEV
     .        ST,                    ! effective thermal source level    NUCSLEV
     .        XNEUT ( WH_TNO ),      ! neutron output, n/KT              F-18
     .        XPGAM ( WH_TNO )       ! prompt gamma energy fraction      F-18


      COMMON /NWCHAR/ EFTHYLD, GAMDT, GDOSE, NWCYLD, SG, SN,
     .                ST, XNEUT, XPGAM
```

```
*
* * ---------------------------------------------------------------------- *
* * PARAM COMMON BLOCK                               04/23/86               *
* * ---------------------------------------------------------------------- *
* *                                                                         *
*       subscripts that are in common with two or more COMMON blocks       **
*
*

      PARAMETER  BL_IDS = 60,        ! blue unit ID   ( ship )

     .           NC_ENV =  8,        ! nuclear environment

     .           NJ_BLU = 6,         ! number of jammers per blue aircraft

     .           NJ_RAC = 12,        ! number of jammers per red aircraft

     .           RAC_LB = 30,        ! red aircraft airframe label

     .           RD_IDM =  30,       ! red group ( or track ) ID number

     .           RD_BOF = 180,       ! red ID for a bomber or fighter

     .           RD_ID =  510,       ! red unit ID

     .           RD_MSL = RD_ID - RD_BOF,
                                     ! red ID - RUACCT
                                     !  (red missile index)

     .           RF_BND =  4,        ! radar frequency bands
                                     ! 1 = A-band
                                     ! 2 = S-band
                                     ! 3 = I-band
                                     ! 4 = J-band

     .           STA_ID = 120,       ! station identifier, for fixed
                                     ! defensive posture, or for response
                                     ! to tactical situation.

     .           STA_DF = 30,        ! station defined for fixed defense
                                     !  posture  ( I = RAC_LB + 1 to
                                     !  AL_PSG - 1)

     .           AL_PSG = RD_IDM + STA_DF + 1,
                                     ! total pseudo groups for coverage,
                                     ! red groups and fixed stations for CAP
                                     ! AL_PSG is used for early warning
                                     ! responsive stationing
```

```
.        VFC_PT = 10,          ! blue fighter platform type
                               ! Note: this number must be greater
                               ! than the number of early warning and
                               ! electronic warfare platform types

.        VQC_PT = 5,           ! blue electronic warfare platform type
                               ! Note: this number must be less than
                               ! the number of fighter platform types

.        VWC_PT = 5,           ! blue early warning platform type
                               ! Note: this number must be less than
                               ! the number of fighter platform types

.        VF_ID  = 120,         ! VF ID number

.        VQ_ID  = 10,          ! VQ ID number

.        VW_ID  = 10,          ! VW ID number

.        AP_ID  = VF_ID + VQ_ID + VW_ID,
                               ! AP ID number

.        BL_ID = BL_IDS + AP_ID,
                               ! blue unit ID   ( any type )

.        AP_RD = AP_ID * RD_ID,
                               ! the total possible number of blue
                               ! aircraft, red unit combinations

.        BL_RD = BL_ID * RD_ID,
                               ! the total possible number of blue
                               ! unit, red unit combinations

.        VF_RD = VF_ID * RD_ID,
                               ! the total possible number of fighter,
                               ! red unit combinations

.        BL_IDC = BL_IDS + VW_ID,
                               ! the total possible number of blue
                               ! aircraft control units

.        AC_ID = VFC_PT + VQC_PT + VWC_PT,
                               ! the possible total number of blue
                               ! aircraft platform types

.        RD_MST = 10           ! red missile type
```

```
*                                                                      *
* -------------------------------------------------------------------- *
*    PROGRM COMMON BLOCK                    09/15/86                   *
* -------------------------------------------------------------------- *
*                           miscellaneous                              *

     PARAMETER LUNINP = 50      ! logical unit number for the input
                                ! files of files

     PARAMETER LUNSUR = 52      ! logical unit number for the input
                                ! files of surveillance messages

     PARAMETER LUNOUT =  6      ! logical unit number for the output file

     PARAMETER LUNMIS = 45      ! logical unit number for the RED
                                ! missile unformatted output file

     PARAMETER LUNRED = 46      ! logical unit number for the RED
                                ! aircraft unformatted output file

     PARAMETER LUNBLU = 47      ! logical unit number for the BLUE
                                ! fighter aircraft unformatted output file

     PARAMETER LUNVAW = 48      ! logical unit number for the BLUE
                                ! early warning aircraft unformatted output file

     PARAMETER LUNSHP = 49      ! logical unit number for the BLUE
                                ! ship damage output file

     PARAMETER LUNVAQ = 51      ! logical unit number for the BLUE
                                ! electronic support aircraft unformatted
                                ! output file

     PARAMETER LUNERR = 99      ! logical unit number for the system error
                                ! messages

*                                    CONTENT                  INITIAL VALUE
*                                                                OR FILE

     REAL*4

    .          BRGREF,          ! Orientation of the NADS X-axis    F-01
                                ! from North

    .          LATREF,          ! Latitude of ZZ, NADS reference    F-01
                                ! point
```

```
     .          LONREF,       ! Latitude of ZZ, NADS reference    F-01
                              ! point

     .          TSTART        ! Time reference for start of NADS  F-01
                              ! simulation in seconds start of the
                              ! decade

     INTEGER*4  IPRINT,       ! Print option for detailed         F-01
                              ! simulation printout

     .          ISEED,        ! Initial random number seed        F-01

     .          JPRINT,       ! Print option for initialization   F-01
                              ! files

     .          TITLE(20),    ! 80-character run title            F-01

     .          IVERSN(3)     ! NADS version number and date   5.2 10/01/85


     INTEGER*2

     .          RANDX,        ! random number seed  (UPPER HALF)   ---

     .          RANDY,        ! random number seed (LOWER HALF)    ---

     .          ASENSR        !  sensor copies created ( 1 =  all; F-01
                              !  0 = all except ship/aircraft)

     LOGICAL*1

     .          PRTOPT(36),   !  print option switch vector for    F-01
                              !  detailed printouts
                              ! index    meaning if TRUE
                              !  0  - error messages always printed
                              !  1  - COMAND events
                              !  2  - COMAND decisions
                              !  3  - COMAND program trace
                              !  4  - COMAND internal subroutine output
                              !  5  - Intercept control events
                              !  6  - Intercept control decisions
                              !  7  - Intercept control program trace
                              !  8  - Intercept control internal
                              !       subroutine output
                              !  9  - Detection events
                              !  10 - Detection decisions
                              !  11 - Detection program trace
                              !  12 - Detection internal
```

```
                                        !       subroutine output
                                        !  13 - RED events
                                        !  14 - RED decisions
                                        !  15 - RED program trace
                                        !  16 - RED internal subroutine output
                                        !  17 - Fighter events
                                        !  18 - Fighter decisions
                                        !  19 - Fighter program trace
                                        !  20 - Fighter internal subroutine output
                                        !  21 - AEW, VAQ events
                                        !  22 - AEW, VAQ decisions
                                        !  23 - AEW, VAQ program trace
                                        !  24 - AEW, VAQ internal subroutine
                                        !       output
                                        !  25 - SAM Ship events
                                        !  26 - SAM Ship decisions
                                        !  27 - SAM Ship program trace
                                        !  28 - SAM Ship internal subroutine
                                        !       output
                                        !  29 - NUCLER events
                                        !  30 - NUCLER decisions
                                        !  31 - NUCLER program trace
                                        !  32 - NUCLER internal subroutine
                                        !       output
                                        !  33 - Damage events
                                        !  34 - Damage decisions
                                        !  35 - Damage program trace
                                        !  36 - Damage internal subroutine
                                        !       output

     .          JAMMING                 !  switch to disable the RED jamming F-01
                                        !  scenario                         .TRUE.

      BYTE      XXXXXX(4)

      EQUIVALENCE(XXXXXX(1),ISEED),(XXXXXX(1),RANDX),(XXXXXX(3),RANDY)


      COMMON /PROGRM/ BRGREF, LATREF, LONREF, TSTART, IPRINT, ISEED,
     .                JPRINT, TITLE, IVERSN, ASENSR, PRTOPT,JAMMING
```

```
*                                                                          *
* ---------------------------------------------------------------------- *
*     RACHAR COMMON BLOCK                        08/23/85                  *
* ---------------------------------------------------------------------- *
*                        red aircraft characteristics                      *

*                                           CONTENT              INITIAL VALUE
*                                                                   OR FILE


**    PARAMETER  RAC_LB = 20,        ! RED aircraft platform type           **
**                                                                          **
**    .          NC_ENV =  8,        ! nuclear environment                  **
**                                                                          **
**    .          NJ_RAC = 12         ! number of jammers per RED aircraft   **
**    PARAMETER  RF_BND =  4         ! radar frequency bands                **
**                                   ! 1 = A-band                           **
**                                   ! 2 = S-band                           **
**                                   ! 3 = I-band                           **
**                                   ! 4 = J-band                           **
**                                                                          **
      PARAMETER  DL_RAC =  2         ! damage level                         **


      REAL*4  RACNED ( RAC_LB ),     ! drift navigation error, type 2     F-16

    .         RACNEH ( RAC_LB ),     ! heading navigation error, type 1   F-16

    .         RACNEV ( RAC_LB ),     ! velocity navigation error, type 1  F-16

    .         RACRCS ( RAC_LB, RF_BND ),
                                     ! radar cross section                F-16

    .         RACRSP ( RAC_LB, NC_ENV, DL_RAC )
                                     ! thresholds of aircraft response    F-36
                                     !  to nuclear environment  NC_ENV
                                     !  is not applicable to type
                                     !  RAC_LB:
                                     !     DL_RAC = 1 is loss of mission
                                     !            capability
                                     !     DL_RAC = 2 is loss of aircraft


      INTEGER*2  RACMTO ( RAC_LB ),  ! total missiles carried             F-16
```

```
     .          RACMTY ( RAC_LB, 4 ),
                                    ! missile type for up to four        F-16
                                    !  missiles, in the order of
                                    !  launching

     .          RACDBL ( RAC_LB ),  ! delay between missile launches,    F-16
                                    !  seconds

     .          RACDLL ( RAC_LB ),  ! delay after last launch            F-16
                                    !  ( for guidance )

     .          RACJTY ( RAC_LB, NJ_RAC ),
                                    ! jammer type for up to NJ_RAC       F-16
                                    ! jammers

     .          RACRSN ( RAC_LB)    ! red sensor number, 1-99            F-16

      LOGICAL*1 RACNER ( RAC_LB)    ! Type of navigation error
                                    ! True - type 1 - heading and/or
                                    !        velocity error
                                    ! False - type 2 - drift error


      CHARACTER*30 RACANM ( RAC_LB ) ! red aircraft name                 F-16


      COMMON /RACHAR/ RACANM, RACRCS, RACRSP, RACMTO, RACMTY, RACDBL,
     .                RACDLL, RACJTY, RACNED, RACNEH, RACNEV, RACNER,
     .                RACRSN
```

```
*
*
* ------------------------------------------------------------------------ *
*    RDUNIT COMMON BLOCK                         08/31/85                  *
* ------------------------------------------------------------------------ *
*                          status of red units                             *

*                                              CONTENT          INITIAL VALUE
*                                                                  OR FILE


**     PARAMETER  RD_ID = 510,          ! ID of any red unit                **
**                                      !  ( aircraft or missile )          **
**                                                                          **
**    .           RD_BOF =160,          ! ID of a red aircraft              **
**                                                                          **
**    .           NJ_RAC = 12           ! number of jammers per red aircraft  **

       PARAMETER TJ_RAC = NJ_RAC * RD_BOF
                                        ! total possible number of RED
                                        ! airborne jammers.


       REAL*4  RUXC ( RD_ID ),          ! x-coordinate              F-23,25

      .        RUYC ( RD_ID ),          ! y-coordinate              F-23,25

      .        RUZC ( RD_ID ),          ! z-coordinate              F-23,25

      .        RUXS ( RD_ID ),          ! x-speed                   F-23,25

      .        RUYS ( RD_ID ),          ! y-speed                   F-23,25

      .        RUZS ( RD_ID ).          ! z-speed                   F-23,25

      .        RUJWMH ( RD_BOF , NJ_RAC)
                                        ! power density of jammer,  F-23,25
                                        !  W/MHz


       INTEGER*2  RUFTY ( RD_ID ),      ! functional type           F-23,25
                                        !  ( negative = killed,
                                        !    0 = not in game,
                                        !    1 = bomber,
                                        !    2 = fighter,
                                        !    3 = recce,
```

```
                                  !    4 = SOJ,
                                  !    5 = SSM,
                                  !    6 = ASM )

    .       RULEAD ( RD_BOF ),    ! ID of the RED leader          F-24

    .       RUPSEN ( RD_BOF ),    ! air search radar sensor type  F-16

    .       RUPTY ( RD_ID ),      ! platform type or              F-23,25
                                  !  missile type

    .       RUTLU ( RD_ID ),      ! time of last update           F-23,25

    .       RUNODE ( RD_ID ),     ! next node of flight plan         1

    .       RUTARG ( RD_ID ),     ! blue ID of target                0
                                  !  ( FTY = 5 or 6 )

    .       RUIDL,                ! last red ID number to enter      0
                                  !  game, so far

    .       RUTGAP ( RD_ID ),     ! time in gap

    .       RUNUCT ( RD_ID ),     ! time that maximum damage         0
                                  !  occurred

    .       RUDMG ( RD_ID ),      ! level of maximum damage          0

    .       RUNUCJ ( RD_ID ),     ! nuclear environment that         0
                                  !  caused maximum damage
                                  !  ( zero is nonnuclear )

    .       RUGID ( RD_BOF ),     ! ID of red group with which       0
                                  !  the aircraft is associated
                                  !  ( zero is no association )

    .       RUJAM ( RD_BOF , NJ_RAC ),
                                  ! blue sensor type targeted by     0
                                  !  jammers 1 to NJ_RAC;
                                  !  ( zero is jammer OFF )

    .       RUACCT,               ! number of aircraft in the red   F-23
                                  !  aircraft scenario

    .       RUTDNV ( RD_ID )      ! time of next change in velocity
                                  !  vector, for updating detection
                                  !  times

      LOGICAL*1
```

```
     .          RUTURN( RD_BOF)      ! Flag to indicate this RED        .FALSE.
                                     !  aircraft is making a turn over
                                     !  30 degrees


        COMMON /RDUNIT/ RULEAD, RUPSEN, RUXC, RUYC, RUZC, RUXS, RUYS,
     .                  RUZS, RUJWMH, RUFTY, RUPTY, RUTLU,
     .                  RUNODE, RUTARG, RUIDL, RUTGAP, RUNUCT,
     .                  RUDMG, RUNUCJ, RUGID, RUJAM, RUACCT,
     .                  RUTDNV, RUTURN
```

```
*                                                                        *
* ---------------------------------------------------------------------- *
*    REPRT COMMON BLOCK                              03/06/86            *
* ---------------------------------------------------------------------- *
*                          status reports summary                        *

*                                              CONTENT          INITIAL VALUE
*                                                                  OR FILE


**    PARAMETER  RD_BOF = 160,        ! red ID  ( aircraft )                **
**                                                                          **
**    .          RD_ID  = 510,        ! red ID  ( any red unit )            **
**                                                                          **
**    .          BL_IDM =  60,        ! blue unit ID  ( any ship unit )     **
**                                                                          **
**    .          VF_ID  =  67         ! VFID  ( = blue unit ID - VFBIAS )   **


      INTEGER*2  RASM ( RD_BOF ),     ! number of ASMs launched             0

      .          RUFTYP ( RD_ID ),    ! red functional type
                                      !   ( 1 = bomber,
                                      !     2 = fighter,
                                      !     3 = recce,
                                      !     4 = SOJ,
                                      !     5 = SSM,
                                      !     6 = ASM )

      .          BSAMC ( BL_IDS ),    ! number of conventional SAMs fired   0

      .          BSAMN ( BL_IDS ),    ! number of nuclear SAMs fired        0

      .          BSAMCD ( BL_IDS ),   ! number of conventional SAMs damaged 0

      .          BSAMND ( BL_IDS ),   ! number of nuclear SAMs damaged      0

      .          VENG ( VF_ID ),      ! number of targets engaged by VF     0

      .          VAIM ( VF_ID, 4 ),   ! number of AIMs fired, types 1 to 4  0

      .          ALNCH ( AP_ID ),     ! number of flights during the game   0

      .          RKILLV ( RD_ID ),    ! blue ID of killing VF               0

      .          RKILLS ( RD_ID ),    ! blue ID of killing ship             0
```

```
.              RPTDEF ( RD_ID ),    ! blue ID of ship whose point          0
                                    !  defense was penetrated

.              RCHIT ( RD_ID ),     ! blue ID of ship hit by               0
                                    !  conventional missile
                                    ! if less than zero, indicates
                                    ! disposition of missile as follows:
                                    !   1 - clobber ( flew into water)
                                    !   2 - dud ( never armed)
                                    !   3 - inflight failure
                                    !   4 - launch abort

.              RENGSH ( RD_ID ),    ! number of BLUE SAMs fired            0
                                    ! against this RED

.              RNHIT ( RD_ID )      ! blue ID of ship targeted by          0
                                    !  nuclear burst


  LOGICAL*1    RENGVF ( RD_ID ),    ! T = engaged by a VF                FALSE

.              VKILLF ( VF_ID )     ! T = killed in a dogfight           FALSE


  COMMON /REPRT/ RASM, RUFTYP, BSAMC, BSAMN, BSAMCD, BSAMND, VENG,
.                VAIM, ALNCH, RKILLV, RKILLS, RPTDEF,
.                RCHIT, RNHIT, RENGVF, RENGSH, VKILLF
```

```
*                                                                        *
* ---------------------------------------------------------------------- *
*     RJCHAR COMMON BLOCK                       08/31/85                 *
* ---------------------------------------------------------------------- *
*                         red jammer characteristics                     *

*                                          CONTENT             INITIAL VALUE
*                                                                 OR FILE


      PARAMETER  RD_JMT = 25


      REAL*4  RJCPOW ( RD_JMT ),     ! total effective radiated         F-27
                                     !  jamming power, watts

     .        RJCMXW ( RD_JMT ),     ! maximum bandwidth, MHz           F-27

     .        RJCMNW ( RD_JMT )      ! minimum bandwidth, MHz           F-27

      INTEGER*2 RJBSN( RD_JMT, 6),   ! RED jamming BLUE Sensor number   F-27

     .          RJREST               ! RED jamming response Time after  F-27
                                     ! receiving signal.

      LOGICAL*1 RJAUTO               ! Switch to control the response of F-27
                                     ! RED aircraft to BLUE tracking.
                                     ! TRUE = aircraft will initiate
                                     ! jamming if possible when tracked.

      COMMON /RJCHAR/ RJBSN, RJCPOW, RJCMXW, RJCMNW, RJREST,
     .                RJAUTO
```

```
*                                                                          *
* ------------------------------------------------------------------------ *
*    RMCHAR COMMON BLOCK                             04/23/86              *
* ------------------------------------------------------------------------ *
*                        red missile characteristics                       *

*                                    CONTENT                  INITIAL VALUE
*                                                                OR FILE


      PARAMETER  ALT_HL =  2         ! high/low altitude
      PARAMETER  DL_RMS =  3         ! damage level

**    PARAMETER  RD_MST = 10         ! red missile type
**    PARAMETER  NC_ENV =  8         ! nuclear environment                **
**    PARAMETER  RF_BND =  4         ! radar frequency bands              **
**                                   ! 1 = C-band                         **
**                                   ! 2 = E/F-band                        **
**                                   ! 3 = I/J-band                        **
**                                   ! 4 = K-band                         **
**                                                                        **


      REAL*4

     .        RMCRMX ( RD_MST ),     ! maximum range                  F-17

     .        RMCRMN ( RD_MST ),     ! minimum range                  F-17

     .        RMCDC ( RD_MST ),      ! downrange distance associated  F-17
                                     !  with launch climb/dive phase

     .        RMCVC ( RD_MST ),      ! launch climb/dive speed        F-17

     .        RMCHA ( RD_MST ),      ! altitude of cruise phase A     F-17

     .        RMCVA ( RD_MST ),      ! speed of cruise phase A        F-17

     .        RMCDD ( RD_MST ),      ! downrange distance associated  F-17
                                     !  with dive/climb between cruise
                                     !  phases A and B

     .        RMCVD ( RD_MST ),      ! speed of dive/climb between    F-17
                                     !  cruise phases A and B

     .        RMCHB ( RD_MST ),      ! altitude of cruise phase B     F-17
```

```
     .       RMCVB ( RD_MST ),       ! speed of cruise phase B            F-17

     .       RMCDT ( RD_MST ),       ! downrange distance associated      F-17
                                     !  with dive/climb on target

     .       RMCVT ( RD_MST )        ! speed of dive/climb on target      F-17

     REAL*4  RMCRNG ( RD_MST ),      ! maximum downrange distance for     F-17
                                     !  start of cruise phase B

     .       RMCRCS ( RD_MST, ALT_HL, RF_BND ),
                                     ! radar cross section                F-17
                                     !
                                     !   ALT_HL = 1, high altitude
                                     !
                                     !   ALT_HL = 2, low altitude

     .       RMCPSF ( RD_MST ),      ! probability of salvage fuze        F-17
                                     !  firing, given a hit by a
                                     !  blue weapon

     .       RMCRSP ( RD_MST, NC_ENV, DL_RMS ),
                                     ! thresholds of vulnerability to     F-37
                                     !  nuclear environments
                                     !
                                     !   DL_RMS = 1, missile airframe or
                                     !          guidance system is
                                     !          damaged; missile killed
                                     !     if nonnuclear
                                     !
                                     !   DL_RMS = 2, salvage fuze is fired
                                     !
                                     !   DL_RMS = 3, warhead disabled
                                     !
                                     !   zero for DL_RMS = 1 makes NC_ENV
                                     !             non-applicable
                                     !
                                     !   1.E60 for DL_RMS = 2 precludes
                                     !             salvage fuzing

     .       RMCHOB ( RD_MST ),      ! planned height of burst; zero      F-17
                                     !  for nonuclear

     .       RMCPWH ( RD_MST ),      ! probability that nuclear warhead   F-17
                                     !  or fuze was killed, given that
                                     !  the missile itself suffered a
                                     !  hard kill without firing salvage
                                     !  fuze
```

```
     .         RMCDUD ( RD_MST ),      ! probability that the missile will  F-17
                                       !  be a dud (that is, probability
                                       !  of launch abort)

     .         RMCARP ( RD_MST ),      ! parameter corresponding to the     F-17
                                       !  indicated missile arming
                                       !  specification (see RMCARM, below)

     .         RMCMTF ( RD_MST ),      ! mean time to failure for the       F-17
                                       !  duration of the missile's flight,
                                       !  assumed to follow an exponential
                                       !  distribution.  added to time of
                                       !  launch to indicate time of
                                       !  missile failure.

     .         RMCMTC ( RD_MST )       ! mean time to failure for low       F-17
                                       !  altitude flight of missile,
                                       !  assumed to follow an exponential
                                       !  distribution.  added to time at
                                       !  which missile first reaches low
                                       !  cruise altitude (low altitude is
                                       !  less than 100. meters).

      INTEGER*2  RMCWH ( RD_MST ),     ! warhead type ( zero is             F-17
                                       !  conventional; positive is index
                                       !  to NWCHAR tables )

     .            RMCARM ( RD_MST )    ! missile arming specification.      F-17
                                       !  indicates that arming parameter
                                       !  specifies (see Figure 12):
                                       !    1 = time offset from node 1;
                                       !    2 = time offset from node 2;
                                       !    3 = time offset from node 3;
                                       !    4 = time offset from node 4;
                                       !    5 = time offset from node 5;
                                       !    6 = time offset from node 6;
                                       !    7 = distance downrange from
                                       !          launch;
                                       !    8 = distance downrange prior
                                       !          to reaching target.


      COMMON /RMCHAR/ RMCRMX, RMCRMN, RMCDC, RMCVC, RMCHA, RMCVA,
     .                RMCDD, RMCVD, RMCHB, RMCVB, RMCDT, RMCVT,
     .                RMCRNG, RMCRCS, RMCPSF, RMCRSP, RMCHOB, RMCPWH,
     .                RMCDUD, RMCARP, RMCMTF, RMCMTC, RMCWH, RMCARM
```

```
*                                                                      *
* -------------------------------------------------------------------- *
*    RSA COMMON BLOCK                          08/27/84                *
* -------------------------------------------------------------------- *
*                            red scenario, aircraft                    *

*                                          CONTENT          INITIAL VALUE
*                                                              OR FILE


**    PARAMETER  RD_BOF =  160,       ! red ID ( for a red aircraft )       **

**    .          NJ_RAC =  12         ! number of jammers per red aircraft  **

      PARAMETER  RD_MSN =   4,       ! red aircraft's missile number

      .          N_ACFP =  30        ! node of aircraft flight path


      REAL*4  RSAXC ( RD_BOF, N_ACFP ), !                              F-23
                                     ! x-coordinate of next way point.

      .       RSAYC ( RD_BOF, N_ACFP ), !                              F-23
                                     ! y-coordinate of next way point.

      .       RSAZC ( RD_BOF, N_ACFP ), !                              F-23
                                     ! z-coordinate of next way point.

      .       RSAV ( RD_BOF, N_ACFP )  !                               F-23
                                     ! velocity to next way point.


      INTEGER*2  RSATRG ( RD_BOF, RD_MSN ), !                          F-23
                                     ! blue ID of target of Mth missile

      .          RSANCT ( RD_BOF ),  ! count of nodes specified        F-23

      .          RSANTY ( RD_BOF, N_ACFP ), !                          F-23
                                     ! node type
                                     !  1 = enter game
                                     !  2 = change in velocity vector
                                     !  3 = radar ON
                                     !  4 = radar OFF
                                     !  5 = red jammer ON
                                     !  6 = red jammer OFF
```

```
                                     !  7 = communications jammer ON
                                     !  8 = communications jammer OFF
                                     !  9 = bomber arrives at launch time
                                     !  10 = leaves game

     .          RSAJSN ( RD_BOF, N_ACFP, NJ_RAC)!
F-23
                                     ! blue sensor type targeted by
                                     !  jammers 1 to NJ_RAC in node type 5


     COMMON /RSA/ RSAXC, RSAYC, RSAZC, RSAV, RSATRG,
     .            RSANCT, RSANTY, RSAJSN
```

```
*                                                                       *
* --------------------------------------------------------------------- *
*     RSCHAR COMMON BLOCK                          02/26/85             *
* --------------------------------------------------------------------- *
*                            sensor characteristics                     *

*                                                CONTENT     INITIAL VALUE
*                                                                 OR FILE


      PARAMETER  RSN_TY =  30        ! Red sensor type


      REAL*4  RSCADD ( RSN_TY),      ! automatic detection delay time  F-31

     .        RSCDR ( RSN_TY ),      ! range beta  ( clear             F-31
                                     !  environment detection
                                     !  range on 1 sq meter target )

     .        RSCSS ( RSN_TY, 3 ),   ! radar search sector descriptors,
                                     ! 1 - fraction of circle covered in
                                     !     radians                     F-31
                                     ! 2 - cosine of half sector       F-31
                                     ! 3 - cosine squared of half sector
                                     !                                 F-31

     .        RSCIR ( RSN_TY ),      ! instrumented range              F-31

     .        RSCDRA ( RSN_TY ),     ! range alpha ( burnthrough       F-31
                                     !  range on 1 sq meter target
                                     !  with SSJ of 1 watt per MHz )
                                     !  squared

     .        RSCN2 ( RSN_TY),       !                                 F-31

     .        RSCSLC ( RSN_TY ),     ! Side lobes jammer suppression   F-31
                                     !  channel (channel, real)

     .        RSCSLR ( RSN_TY ),     ! sidelobe ratio  ( average       F-31
                                     !  arithmetic ratio of main
                                     !  beam to sidelobes )

     .        RSCML ( RSN_TY),        ! main lobe size ( half width of  F-31
                                     !  main lobe vulnerable to main
                                     !  lobe jamming )
```

```
     .        RSCTBS ( RSN_TY),       !  detection delay base time      F-31

     .        RSCJDW ( RSN_TY ),      ! jamming bandwidth               F-31

     .        RSDPTH ( RSN_TY )       ! doppler velocity threshold for  F-31
                                      ! detection (used to select
                                      ! alternate radar mode)


      INTEGER*2  RSALTM ( RSN_TY )    ! alternate radar mode index


      COMMON /RSCHAR/ RSCADD, RSCDR, RSCSS, RSCIR, RSCDRA, RSCSLC,
     .                RSCSLR, RSCML, RSCJDW, RSDPTH, RSALTM, RSCN2,
     .                RSCTBS
```

```
*                                                                        *
* ---------------------------------------------------------------------- *
*     RSM COMMON BLOCK                          09/03/85                 *
* ---------------------------------------------------------------------- *
*                             red scenario, missiles                     *

*                                            CONTENT          INITIAL VALUE
*                                                                OR FILE


**    PARAMETER  RD_MSL =  RD_ID - RD_BOF                                  **
                                   ! red ID - RUACCT
                                   !  (red missile index)

      PARAMETER  N_MSFP =    7     ! next node of missile flight
                                   !  plan

      PARAMETER  RMS_FP = RD_MSL * N_MSFP

      REAL*4  RSMXS ( RD_MSL, N_MSFP ), !                                0.
                                   ! x velocity to next way-point.

    .         RSMYS ( RD_MSL, N_MSFP ), !                                0.
                                   ! y velocity to next way-point.

    .         RSMZS ( RD_MSL, N_MSFP ), !                                0.
                                   ! z velocity to next way-point.

    .         RSMTNN( RD_MSL, N_MSFP)    !
                                   !  time to reach next node


      INTEGER*2  RSMNTY( RD_MSL, N_MSFP ),!                              9
                                   ! event type:
                                   !   0 = launch;
                                   !   1 = reach cruise A altitude;
                                   !   2 = mid-flight dive;
                                   !   3 = reach cruise B altitude;
                                   !   4 = start terminal dive;
                                   !   5 = reach target aimpoint;
                                   !   7 = arm missile;
                                   !   8 = nuclear detonation;
                                   !   9 = terminate missile.

    .            RSMPTY            ! Number of RED missiles defined.   F-17
```

```
LOGICAL*1  RSMARM ( RD_MSL )   ! missile is armed            .FALSE.

COMMON /RSM/ RSMXS, RSMYS, RSMZS,  RSMTNN, RSMNTY, RSMPTY, RSMARM
```

```
*                                                                        *
* ---------------------------------------------------------------------- *
*     SCALT COMMON BLOCK                          12/09/83               *
* ---------------------------------------------------------------------- *
*             scaling factors for 1-KT sea level nuclear bursts          *

*                                              CONTENT          INITIAL VALUE
*                                                                  OR FILE


      REAL*4  SIMPUL,                 !  impulse scale factor

     .        SPRESS,                 !  pressure scale factor

     .        STIME,                  !  time scale factor

     .        SDIST,                  !  distance scale factor

     .        YSCL                    !  scaled yield
                                      !   ( cube root of KT )


      COMMON /SCALT/ SIMPUL, SPRESS, STIME,
     .                SDIST, YSCL
```

```
*                                                                        *
* ---------------------------------------------------------------------- *
*     SHCHAR COMMON BLOCK                              05/16/85          *
* ---------------------------------------------------------------------- *
*                          ship characteristics                          *

*                                               CONTENT        INITIAL VALUE
*                                                                 OR FILE


      PARAMETER  SH_PUT = 10,       ! ship platform type  ( class )

    .            DL_SHC =  6        ! damage level                       **

**    PARAMETER  NC_ENV =  8,       ! nuclear environment                **
**                                                                       **


      REAL*4  SHCCJT ( SH_PUT ),    ! communication jamming threshold    ---

    .         SHCCTP ( SH_PUT ),    ! communication transmitter power    ---


    .         SHCRAC ( SH_PUT ),    ! SAM action range                   ---

    .         SHCHCP ( SH_PUT ),    ! hot CPA ( implies threat to own    F-14
                                    !  ship )

    .         SHCRSP ( SH_PUT , NC_ENV , DL_SHC ),
                                    ! nuclear damage threshold levels,   F-34
                                    !  for response to each nuclear
                                    !  environment
                                    !
                                    !    DL_SHC = 1 is radar down,
                                    !
                                    !    DL_SHC = 2 is fifty per cent
                                    !           weapon delivery impairment
                                    !
                                    !    DL_SHC = 3 is one hundred per
                                    !             cent weapon delivery
                                    !             impairment
                                    !
                                    !    DL_SHC = 4 is ninety per cent
                                    !                mobility impairment
                                    !
                                    !    DL_SHC = 5 is ninety per cent
                                    !             seaworthiness impairment
```

```
                                      !
                                      !   DL_SHC = 6 is ship destroyed
                                      !               (sunk)

  .       SHCDPD ( SH_PUT ),          ! action radius of point defense     F-14
                                      !  hemisphere

  .       SHCPKP ( SH_PUT , 2 ),      ! point defense system kill          F-14
                                      !  probability against sea skimmer
                                      !  and diving type missiles

  .       SHCECM ( SH_PUT ),          ! fraction of point defense kills    F-14
                                      !  attributable to ECM

  .       SHRXSC ( SH_PUT ),          ! radar cross-section of red         F-14
                                      ! aircraft (square meters)

  .       SHSENH ( SH_PUT )           ! initial sensor height (feet)       F-14

   INTEGER*2  SHCMTY ( SH_PUT , 2 ),
                                      ! SAM types  ( conv and nuc )        F-14

  .           SHCSTY ( SH_PUT , 4 ),
                                      ! sensor types of 1 to 4 search      F-14
                                      !  radars

  .           SHCADT ( SH_PUT ),      ! total tracking capacity of         F-14
                                      !  search radars

  .           SHCFCS ( SH_PUT ),      ! type of fire control system        F-14

  .           SHCLTP ( SH_PUT ),      ! SAM launcher type                  F-14
                                      !  pointer into the BLUE launcher
                                      !  characteristics arrays, BLCHAR.CMN

  .           SHCNOC ( SH_PUT ),      ! number of fire control channels    F-14
                                      ! operable

  .           SHCNOL ( SH_PUT ),      ! number of SAM launchers operable   F-14

  .           SHCHIT ( SH_PUT,2),     ! number of hits required by         F-14
                                      !  conventional antiship missiles
                                      !  to cause fifty and one hundred
                                      !  percent weapon delivery impairment

  .           SHCVC ( SH_PUT ),       ! max number of VF that can be       F-14
                                      ! vectored to the controller
```

```
     .          SHCVF ( SH_PUT ),    ! max number of VF that can be       F-14
                                     ! assigned to the controller

     .          SHJTYP ( SH_PUT, 2 ),
                                     ! jammer type, (integer from         F-14
                                     ! 1 to 8)

     .          SHJNUM ( SH_PUT, 2 )
                                     ! jammer number, corresponding to    F-14
                                     ! a jammer type.


      LOGICAL*1 SHCTDS ( SH_PUT ),   ! tactical data system capability,   F-14
                                     !  T or F

     .          SHCCC ( SH_PUT )     ! command center capability, T or F  F-14


      COMMON /SHCHAR/ SHCCJT, SHCCTP, SHCRAC, SHCHCP, SHCRSP, SHCDPD,
     .                SHCPKP, SHCECM, SHCMTY, SHCSTY, SHCADT, SHCLTP,
     .                SHCNOC, SHCNOL, SHCHIT, SHCTDS, SHCCC, SHCFCS,
     .                SHCVC, SHCVF, SHJNUM, SHJTYP, SHRXSC, SHSENH
```

```
*                                                                          *
* ------------------------------------------------------------------------ *
*     SHSTAT COMMON BLOCK                         06/10/86                  *
* ------------------------------------------------------------------------ *
*                              ship status                                 *

*                                               CONTENT         INITIAL VALUE
*                                                                  OR FILE


**    PARAMETER  BL_IDS = 60          ! blue unit ID of ship               **

**    PARAMETER  RD_ID =  510,        ! red unit ID

      PARAMETER  FR_CNC = 24,         ! fire control channel

     .           IL_OGC =  8,         ! illuminator or guidance channel

     .           BL_FRC = BL_IDS * FR_CNC,
                                      ! blue force fire control channels

     .           BL_SRD = BL_IDS * RD_ID * 2,
                                      ! size of the SAM envelope entry
                                      ! and exit data array

     .           BL_ILC = BL_IDS * IL_OGC
                                      ! blue force illuminator or fire
                                      ! control channels



      REAL*4  SSPKPD ( BL_IDS, 2 ),   ! point defense system Pk against    F-21
                                      !  sea-skimmers, and diving type
                                      !  missiles
     .        SSSECT ( BL_IDS, 3 )    ! x ,y and beta for self assignment  F-21
                                      !  calculations

      INTEGER*2  SSCHAV ( BL_IDS ),   ! number of fire control           F-14/21
                                      !  channels free for use

     .           SSCHIT ( BL_IDS ),   ! count of hits on ship by            0
                                      !  conventional warheads

     .           SSCHUP ( BL_IDS ),   ! number of fire control           F-14/21
                                      !  channels in UP status
```

```
.          SSCNT,               ! total number of ships in game      F-21

.          SSFPOL( BL_IDS),     ! The SAM firing policy for          F-21
                                ! this ship.

.          SSFSTA ( BL_IDS , FR_CNC ),
                                ! fire control channel status          0
                                !  ( 0      = available,
                                !    1 - 10 = occupied preemptable
                                !             by higher priority
                                !             target,
                                !    11     = post launch occupied
                                !    12     = not available).

.          SSFTRG ( BL_IDS , FR_CNC ),
                                ! red IDs of targets to which fire     0
                                !  control channels are assigned

.          SSGCUP ( BL_IDS ),   ! number of guidance channels     F-14/21
                                !  in UP status

.          SSLLOD ( BL_IDS, 2 ),
                                ! missiles loaded on launcher
                                !  ( 0 = empty,
                                !    1 = one conv,
                                !    2 = two conv,
                                !    21 = one nuc,
                                !    22 = two nuc )

.          SSLSTA ( BL_IDS, 2 ),
                                ! launcher state                  F-14/21

.          SSLTRG ( BL_IDS, 2 ),
                                ! red ID of target to which launcher   0
                                !  is assigned

.          SSMTOT ( BL_IDS, 2 ),
                                ! total SAMs remaining in         F-14/21
                                !  ship; 1 = conv,   2 = nuc

.          SSNLUP ( BL_IDS ),   ! number of launcher channels     F-14/21
                                !  ( rails ) UP

.          SSITGT ( BL_IDS, IL_OGC ),
                                ! red ID of target to which            0
                                ! illuminator is assigned

.          SSSTC1 ( BL_IDS, RD_ID, 2 ),
```

```
                                    ! Time that RED target enters the   NEVER
                                    ! SAM envelope ( 1 - conventional,
                                    !  2 - nuclear).

     .          SSSTC2 ( BL_IDS, RD_ID, 2 ),
                                    ! Time that RED target leaves the     0
                                    ! SAM envelope ( 1 - conventional,
                                    !  2 - nuclear).

     .          SSTGAV ( BL_IDS, IL_OGC )
                                    ! time that guidance channel ( or     0
                                    !  illuminator ) will be available


      LOGICAL*1  SSNUCD ( BL_IDS )  ! ship has local nuclear weapons    F-21
                                    !  release  ( T or F )


      COMMON/ SHSTAT/ SSSECT, SSPKPD, SSMTOT, SSCHUP, SSCHAV,
     .                SSFSTA, SSFTRG, SSNLUP, SSLSTA, SSLTRG,
     .                SSLLOD, SSGCUP, SSSTC1, SSSTC2, SSTGAV,
     .                SSCHIT, SSCNT, SSITGT, SSFPOL, SSNUCD
```

```
*                                                                           *
* ------------------------------------------------------------------------ *
*     SMCHAR COMMON BLOCK                          04/23/86                 *
* ------------------------------------------------------------------------ *
*                             SAM characteristics                           *


*                                              CONTENT          INITIAL VALUE
*                                                                 OR FILE


      PARAMETER  SAM_TY = 10,          ! SAM type
     .           RD_MTT = RD_MST + 1
                                       ! red missile target type
                                       !  ( T = 11 is any red aircraft )


      REAL*4
     .        SMCEIL ( SAM_TY ),
                                       ! combat ceiling for the SAM          F-15
     .        SMCSPD ( SAM_TY, RD_MTT ),
                                       ! speed   ( average horizontal        F-15
                                       !           component )
     .        SMCMXC ( SAM_TY, RD_MTT ),
                                       ! maximum cross range                 F-15
     .        SMCMNC ( SAM_TY, RD_MTT ),
                                       ! minimum cross range                 F-15
     .        SMCMNR ( SAM_TY, RD_MTT ),
                                       ! minimum range                       F-15
     .        SMCEVA ( SAM_TY, RD_MTT ),
                                       ! A-coefficient for superellipse      F-15
     .        SMCEVB ( SAM_TY, RD_MTT ),
                                       ! B-coefficient for superellipse      F-15
     .        SMCEVN ( SAM_TY, RD_MTT ),
                                       ! exponent for superellipse           F-15
     .        SMCPHT ( SAM_TY, RD_MTT, 3 ),
                                       ! probability of hit table            F-15
```

```
     .        SMCPGP ( SAM_TY, RD_MTT ),
                                    ! probability of hit in gap          F-15

     .        SMCVMX ( SAM_TY, RD_MTT)  ! SAMENV constant


      INTEGER*2  SMCGTP ( SAM_TY ),  ! SAM guidance type                 F-15
                                     !  ( 1 = command all-the-way,
                                     !    2 = home all-the-way,
                                     !    3 = mid-course guidance )

     .          SMCTOF ( SAM_TY, RD_MTT, 3 ),
                                     ! time of flight table              F-15
                                     !  corresponding to P-hit

     .          SMCTGP ( SAM_TY, RD_MTT ),
                                     ! length of time in gap             F-15

     .          SMCWH ( SAM_TY )     ! warhead type ( 0 = conv,          F-15
                                     !  positive is index to NWCHAR )


      COMMON /SMCHAR/ SMCEIL, SMCSPD, SMCMXC, SMCMNC, SMCMNR, SMCEVA,
     .                SMCEVB, SMCEVN, SMCPHT, SMCPGP, SMCVMX, SMCGTP,
     .                SMCTOF, SMCTGP, SMCWH
```

```
*                                                                          *
* ------------------------------------------------------------------------ *
*     SNCHAR COMMON BLOCK                          10/19/85                *
* ------------------------------------------------------------------------ *
*                                sensor characteristics                    *

*                                                CONTENT          INITIAL VALUE
*                                                                   OR FILE


      PARAMETER  SEN_TY =  30,        ! sensor type

     .           SN_SLC =  10         ! half the number of coherent side
                                      ! lobe jammer cancellation channels
                                      ! ( one channel cancels both the
                                      ! direct path and the reflected path
                                      ! jammer)


      REAL*4  SNCADD ( SEN_TY),       ! automatic detection delay time  F-11

     .        SNCDRB ( SEN_TY ),      ! range beta  ( clear             F-11
                                      !  environment detection
                                      !  range on 1 sq meter target )

     .        SNCSS ( SEN_TY ,3),     ! radar search sector descriptors,
                                      ! 1 - fraction of circle covered in
                                      !     radians                     F-11
                                      ! 2 - cosine of half sector       F-11
                                      ! 3 - sine of half sector         F-11

     .        SNCIR ( SEN_TY ),       ! instrumented range              F-11

     .        SNCDRA ( SEN_TY ),      ! range alpha ( burnthrough       F-11
                                      !  range on 1 sq meter target
                                      !  with SSJ of 1 watt per MHz )
                                      !  squared

     .        SNCN2 ( SEN_TY),        ! delay time multiplier           F-11


     .        SNCSLR ( SEN_TY ),      ! sidelobe ratio  ( average       F-11
                                      !  arithmetic ratio of main
                                      !  beam to sidelobes )

     .        SNCMLJ ( SEN_TY, 3 ),
```

```
                                     ! radar vulnerability to       F-11
                                     ! main lobe jamming discriptors
                                     ! 1 - Width of the main lobe
                                     ! 2 - Cosine of the width of
                                     !     the main lobe
                                     ! 3 - Sine of the width of the
                                     !     main lobe
     .       SNCTBS ( SEN_TY ),      ! detection delay base time    F-11

     .       SNCJDW ( SEN_TY ),      ! jamming bandwidth            F-11

     .       SNDPTH ( SEN_TY )       ! doppler velocity threshold for
                                     ! detection (used to select
                                     ! alternate radar mode), in knots


      INTEGER*2 SNALTM ( SEN_TY ),   ! alternate radar mode index

     .         SNCSLC ( SEN_TY ),    ! number of sidelobe jammer    F-11
                                     ! suppression  channel.

     .         SNCFB (SEN_TY )     ! sensor frequency band          F-11
                                     !  1 - C band
                                     !  2 - E/F band
                                     !  3 - I/J band
                                     !  4 - K band

      LOGICAL*1 SNC3D( SEN_TY)      ! indicates the sensor is 3-D   F-11


      COMMON /SNCHAR/ SNCADD, SNCDRB, SNCSS, SNCIR, SNCDRA, SNCMLJ,
     .                SNCSLR, SNCJDW, SNDPTH, SNCN2,
     .                SNCTBS, SNALTM, SNCSLC, SNCFB, SNC3D
```

```
*                                                                        *
* ---------------------------------------------------------------------- *
*     TACTIC COMMON BLOCK                        03/25/86                *
* ---------------------------------------------------------------------- *
*                            miscellaneous                               *

*                                          CONTENT              INITIAL VALUE
*                                                                  OR FILE

     PARAMETER BL_TARTP = 11    ! The total number of missile target
                                ! types and one aircaft target type.

     REAL*4

   .        DDESP,              ! last chance distance for setting   F-19
                                ! nucAIM flag

   .        DISTT2,             ! "good enough" distance for         F-19
                                !  secondary target selection

   .        MININT,             ! minimum permissible distance from  F-19
                                !  force center for VF intercepts

   .        ANGOFF( VFC_PT),    ! angle off jamming strobes for      F-19
                                !  positioning fighters

   .        STARNG( VFC_PT),    ! stationing range from force center F-19
                                !  for positioning fighters against
                                !  jamming strobes

   .        STAANG( VFC_PT),    ! stationing altitude for            F-19
                                !  positioning fighters against
                                !  jamming strobes

   .        PTOGAS( VFC_PT),    ! post-take-off refueling            F-19
                                !  fuel to be transferred to the
                                !  fighter if available

   .        PLNGAS( VFC_PT),    ! quantity of fuel assumed by the    F-19
                                !  command center for planning
                                !  purposes for this type fighter

   .        PTOFTM( VFC_PT),    ! post-take-off refueling            F-19
                                !  time penality if fuel transferred
                                !  to the fighter

   .        LATRNG( VWC_PT),    ! lateral range for airborne early   F-19
                                !  warning aircraft assumed for
```

```
                                ! reactive stationing

.       STADST( VWC_PT),        ! stationing distance for airborne   F-19
                                !  early warning aircraft used for
                                !  reactive stationing ( zero if
                                !  reactive stationing is not used)

.       STAALT( VWC_PT, 2),         !                                F-19
                                ! altitude for stationing airborne
                                !  early warning aircraft
                                !  1 - normal altitude (used for
                                !      reactive stationing to
                                !      surveillance and contact reports
                                !      zero if reactive stationing is
                                !      desired)
                                !  2 - minimum altitude ( used for
                                !      response to jamming)

.       VAWSEC( VWC_PT),        ! cosign of the sector angle         F-19
                                !  covered by early warning
                                !  aircraft ( angle with
                                !  sine[ LATRNG / STADST])

.       NAIMKR,                 ! nuclear AIM expected kill radius   F-19

.       NKORNG( BL_TARTP)       ! Nuclear Keep Out RaNGe for BLue    F-19
                                ! TARget TyPes.
                                ! NOTE: The 11 th position will hold
                                !       the Keep out range for RED
                                !       aircraft


INTEGER*2  COMTDS,              ! message delay time between TDS     F-19
                                !  units

.          COMVCE,              ! message delay if either unit is    F-19
                                !  non-TDS

.          TWAITC,              ! time that command center awaits    F-19
                                !  confirmation of a target
                                !  assignment, before reassigning it

.          TWAITL,              ! time that command center awaits    F-19
                                !  confirmation of a new VF launch
                                !  and target assignment

.          TWAITF,              ! time a  VF awaits response to      F-19
                                !  self-assign request
```

```
        .        TMARG,        ! time margin for DLI launch       F-19
                               !  estimation

        .        MNDOGF,       ! mean duration of a dogfight      F-19

        .        SDDOGF,       ! standard deviation of dogfight   F-19
                               !  duration

        .        NUCNO,        ! number of targets close to       F-19
                               !  primary target, that will set
                               !  nucAIM selection flag

        .        CDOCT,        ! SAM coordination level           F-19
                               !  ( 1 = command center,
                               !    2 = ship sectors,
                               !    3 = no coordination )

        .        TDUO,         ! time between 2 rounds of SAM     F-19
                               !  salvo

        .        TMINS,        ! SAM envelope time threshold,     F-19
                               !  for quick reaction

        .        TEVALS,       ! time to evaluate SAM hit or miss F-19

        .        TOOLT         ! "too late" time for cancel or    F-19
                               !  reschedule A/C launch


   LOGICAL*1     NUCREL,       ! nuclear weapons release for
                               !  force

        .        AUTOESM,      ! ESM bearing ( jamming strobes)
                               !  automatically updated in TDS
                               !   ( T = automatically updated,
                               !     F = contact reports only )

        .        AUTOVAW,      ! program wil station airborne
                               !  early warning aircraft during
                               !  the battle ( T = automatically
                               !  positioned, F = otherwise )

        .        JAMRES,       ! program wil position fighter
                               !  aircraft against jamming
                               !  ( T = automatically
                               !  positioned, F = otherwise )
```

```
     .          SALVO2,        ! VF launch policy for type 2
                               !  missiles
                               !   ( T = two-missile salvo,
                               !     F = single )

     .          SALVO3         ! VF launch policy for type 3
                               !  missiles

      CHARACTER*7
     .          STASPD( VFC_PT)! Speed to be used by fighter in     F-19
                               !responding to jamming


      COMMON /TACTIC/ DDESP, DISTT2, MININT, NAIMKR, NKORNG,
     .         ANGOFF, STARNG, STAANG, LATRNG, PTOGAS, PLNGAS, PTOFTM,
     .         STADST, STAALT, VAWSEC,COMTDS, COMVCE, TWAITC, TWAITL,
     .         TWAITF, TMARG, MNDOGF, SDDOGF, NUCNO, CDOCT, TDUO,
     .         TMINS, TEVALS, TOOLT, AUTOESM, AUTOVAW, JAMRES, NUCREL,
     .         SALVO2, SALVO3, STASPD
```

```
*                                                                       *
* --------------------------------------------------------------------- *
*     TGT COMMON BLOCK                            08/31/85              *
* --------------------------------------------------------------------- *
*                         arrays for TGTCAP and TGTSAM                  *

*                                              CONTENT         INITIAL VALUE
*                                                                 OR FILE

**    PARAMETER  RD_ID = 510         ! number of red units                **

      PARAMETER  RD_VF = RD_ID * VF_ID
                                     ! combination of red and blue units

      REAL*4

     .           AIRMOR( RD_ID),     ! measure of risk for ordering the
                                     ! air list

     .           SAMMOR( RD_ID)      ! measure of risk for ordering the
                                     ! SAM list

      INTEGER*2

     .           AIRLST ( RD_ID ),   ! red IDs of known targets in
                                     !  forward air zone

     .           AIRASN ( RD_ID ),   ! VFID of VF assigned to AIRLST
                                     !  entry

     .           AIRVAW ( RD_ID ),   ! Station for early warning
                                     ! aircraft which covers this RED

     .           AIRCLK ( RD_ID ),   ! time that confirmation of          0
                                     !  assignment is expected
                                     !  ( zero for unassigned targets )

     .           AIRCNT,             ! count of current AIRLST entries    0

     .           JAMLST( RD_ID),     ! RED id's of the current list of
                                     ! active jammers

     .           PILCAP ( RD_VF ),   ! VFID of CAP in primary intercept
                                     !  list  ( PIL )
     .           PILAC  ( RD_VF ),   ! Platform type of CAP in primary
                                     !  intercept list  ( PIL )

     .           PILTGT ( RD_VF ),   ! AIRLST index of target
```

```
     .          PILTIM ( RD_VF ),     ! time of intercept for VF-target
                                      !  pair

     .          PILCNT,               ! count of entries in PIL            0

     .          PILDEL,               ! count of deletions from PIL        0

     .          REDLST( RD_ID),       ! RED id's of the current list of
                                      ! uncovered targets

     .          SAMLST ( RD_ID ),     ! red IDs of all known red missiles,
                                      !  and red airplanes that are inside
                                      !  the minimum intercept range for
                                      !  VF  ( MININT )

     .          SAMCLK ( RD_ID ),     ! estimated time that target will
                                      !  reach force center

     .          SAMCNT                ! number of SAMLST entries

      LOGICAL*1
     .          SAMTGS ( RD_ID )     ! indicator of whether target in      0
                                      !  SAMLST is assigned
                                      !  ( TRUE = assigned,  FALSE = unassigned )
       EQUIVALENCE (JAMLST, REDLST)


      COMMON /TGT/ AIRMOR, SAMMOR, AIRLST, AIRASN, AIRVAW, AIRCLK,
     .             AIRCNT, PILAC, PILCAP, PILTGT, PILTIM, PILCNT,
     .             PILDEL, REDLST, SAMLST, SAMCLK, SAMCNT, SAMTGS
```

```
*                                                                       *
* --------------------------------------------------------------------- *
*    VFCHAR COMMON BLOCK                        02/26/85                *
* --------------------------------------------------------------------- *
*                       Blue fighter( VF) characteristics               *

*                                            CONTENT           INITIAL VALUE
*                                                                 OR FILE

**    PARAMETER  VFC_PT = 10,        ! blue fighter platform type          **
**                                   ! Note: this number must be greater   **
**                                   ! than the number of early warning and **
**                                   ! electronic warfare platform types   **
**                                                                         **
**   .           NC_ENV = 8          ! nuclear environment ( see /ENV/ )   **

**                                                                         **
      PARAMETER  DL_VFC =  2         ! damage level                        **

      REAL*4  VFCSP ( VFC_PT, 4 ),   ! speed ( 1 = max endurance,       F-12
                                     !         2 = max range,
                                     !         3 = buster,
                                     !         4 = gate)
                                     !   meters per second

     .        VFCFBR ( VFC_PT, 4 ),  ! fuel burn rate at four           F-12
                                     !  speeds, kilograms per second

     .        VFCFCR ( VFC_PT, 4 ),  ! fuel consumption rate,           F-12
                                     !  kilograms per meter

     .        VFCFR ( VFC_PT ),      ! reserve fuel                     F-12

     .        VFCFI ( VFC_PT ),      ! fuel planned for intercept       F-12

     .        VFCCJT ( VFC_PT ),     ! comm jamming threshold           F-12

     .        VFCCTP ( VFC_PT ),     ! comm transmitter power           F-12

     .        VFCBTA ( VFC_PT ),     ! aircraft combat ceiling          F-12

     .        VFCER ( VFC_PT ),      ! exchange ratio ( red fighters    F-12
                                     !  killed per blue fighter killed )

     .        VFCRSP ( VFC_PT, NC_ENV, DL_VFC ),
```

```
                                  ! thresholds of aircraft damage     F-32
                                  !  level response to nuclear
                                  !  environments 1 to NC_ENV:
                                  !  a zero entry for (VFC_PT,NC_ENV,1)
                                  !  makes environment NC_ENV not
                                  !  applicable to aircraft type VFC_PT
                                  !
                                  !    DL_VFC = 1 is loss of mission
                                  !          capability
                                  !
                                  !    DL_VFC = 2 is loss of aircraft

     .       VFCLMD ( VFC_PT, 2 ), ! distance travelled during climb  F-12
                                  !  to altitude
                                  !  ( 1 = normal,
                                  !    2 = minimum time to intercept )

     .       VFCLMF ( VFC_PT, 2 ), ! fuel consumed during climb to    F-12
                                  !  altitude, for normal and fast
                                  !  climb profiles

     .       VFCLMA ( VFC_PT, 2 )  ! climbout altitude for the two    F-12
                                  !  profiles


      INTEGER*4  VFCNAM ( VFC_PT, 2 )
                                  ! eight-character name for blue     F-12
                                  !  fighter type


      INTEGER*2  VFCSTY ( VFC_PT ),  ! sensor type                     F-12

     .           VFCADT ( VFC_PT ),  ! tracking capacity               F-12

     .           VFCLMT ( VFC_PT, 2 )
                                  ! time consumed during climb to     F-12
                                  !  altitude, for normal and fast
                                  !  climb profiles


      LOGICAL*1  VFCTDS ( VFC_PT )   ! tactical data system            F-12
                                  !  capability, true or false
```

```
      COMMON /VFCHAR/ VFCSP, VFCFBR, VFCFCR, VFCFR, VFCFI,
     .                VFCBTA, VFCCJT, VFCCTP, VFCER, VFCRSP,
     .                VFCLMD, VFCLMF, VFCLMA, VFCNAM, VFCSTY,
     .                VFCADT, VFCLMT, VFCTDS
```

```
*                                                                          *
* ------------------------------------------------------------------------ *
*     VFSTAT COMMON BLOCK                            08/27/85              *
* ------------------------------------------------------------------------ *
*                               VF status                                  *


*                                              CONTENT         INITIAL VALUE
*                                                                 OR FILE


**    PARAMETER  VF_ID = 67        ! VFID = blue unit ID - VFBIAS           **
**               RD_ID = 510,      ! red unit ID


      REAL*4

    .        VFRINT ( VF_ID , RD_ID),!                               - RNEVER
                                   ! Initial detection of the RED unit
                                   ! by this fighter

    .        VFIXC ( VF_ID ),      ! x-coordinate of planned           ---
                                   !  intercept

    .        VFIYC ( VF_ID ),      ! y-intercept of planned            ---
                                   !  intercept

    .        VFIZC ( VF_ID ),      ! z-intercept of planned            ---
                                   !  intercept

    .        VFIXST,               ! tentative x-velocity              ---

    .        VFIYST,               ! tentative y-velocity              ---

    .        VFIZST                ! tentative z-velocity              ---


      INTEGER*2
    .            VFSSTT ( VF_ID ),  ! tentative speed status           ---

    .            VFSTAT ( VF_ID ),  ! VF state                         ---
```

```
                              ! ( 0 = on deck
                              !   1 = post-launch climbout
                              !   2 = enroute to station
                              !   3 = on station
                              !   4 = pursuing jammer
                              !   5 = vectored intercept
                              !   6 = ready for self-directed intercept
                              !   7 = self-directed intercept
                              !   8 = in Launch Acceptable Region
                              !   9 = ACM with RED escort fighter
                              !  10 = Return to Base )

.   VFSST ( VF_ID ),          ! speed status                      ---
                              ! ( 1 = max endurance,
                              !   2 = max range,
                              !   3 = buster,
                              !   4 = gate )

.   VFMTYP ( VF_ID, 4 ),
                              ! missile types loaded              F-22

.   VFMREM ( VF_ID, 4 ),
                              ! count of missiles remaining

.   VFTARG ( VF_ID ),         ! ID of red target that VF is        0
                              !  assigned to engage

.   VFPNTR( VF_ID ),          !  The pointer to link the APSTAT
                              !  to this VFSTAT common

.   VFCNT,                    ! number of VFs that have       F-21/22
                              !  entered game

.   VFSPTY,                   ! number of fighter types defined   F-12

.   VFMSTY ( VF_ID ),         ! missile type selected for firing  ---

.   VFMSCT ( VF_ID ),         ! number of missiles to be fired    ---

.   VFCPID ( VF_ID ),         ! station ID to which VF is         F-22
                              !  assigned  ( CPSTAT index )

.   VFSEL1 ( VF_ID ),         ! red ID of primary target           0

.   VFSEL2 ( VF_ID ),         ! red ID of secondary target         0

.   VFFRST ( VF_ID ),         ! firing status                      0
                              !  ( 0 = no target,
                              !    1 = reattack,
```

```
                                        !    2 = wait for hit or miss
                                        !        but no reattack )

     .          VFMITC( VF_ID),         ! time of planned intercept of
                                        ! missile fired in home-on-jam mode

     .          VFITC( VF_ID)           ! time of planned intercept)


      LOGICAL*1

     .          VFNUCF ( VF_ID ),       ! nuclear weapon flag ( T = nuc       F
                                        !  is preferred )

     .          VFSELF ( VF_ID ),       ! awaiting response to self-assign    F
                                        !  request

     .          VFNLAR ( RD_ID, VF_ID)
                                        ! lar not achievable against this red


      COMMON /VFSTAT/ VFRINT, VFIXC,  VFIYC,  VFIZC,  VFIXST, VFIYST,
     .                VFIZST, VFITC,  VFMITC, VFSSTT, VFSTAT, VFSST,
     .                VFMTYP, VFMREM, VFTARG, VFCNT, VFSPTY,
     .                VFMSTY, VFMSCT, VFCPID, VFSEL1, VFPNTR,
     .                VFSEL2, VFFRST, VFNUCF, VFSELF, VFNLAR
```

```
*                                                                          *
* ------------------------------------------------------------------------ *
*     VQCHAR COMMON BLOCK                        07/26/84                  *
* ------------------------------------------------------------------------ *
*                 Blue electronic warfare (VAQ) characteristics            *


*                                              CONTENT          INITIAL VALUE
*                                                                  OR FILE


**    PARAMETER  VQC_PT = 5,        ! blue electronic warfare platform type **
**                                  ! Note: this number must be less than   **
**                                  ! the number of fighter platform types  **
**                                                                          **
**    .          NC_ENV = 8         ! nuclear environment ( see /ENV/ )     **

      PARAMETER  DL_VQC =  2        ! damage level                          **


      REAL*4

      .       VQCSP ( VQC_PT, 3 ),  ! speed ( 1 = max endurance,        F-07
                                    !         2 = max range,
                                    !         3 = buster)
                                    !   meters per second

      .       VQCFBR ( VQC_PT, 3 ), ! fuel burn rate at three           F-07
                                    !  speeds, kilograms per second

      .       VQCFCR ( VQC_PT, 3 ), ! fuel consumption rate,            F-07
                                    !  kilograms per meter

      .       VQCFR ( VQC_PT ),     ! reserve fuel                      F-07

      .       VQCCJT ( VQC_PT ),    ! comm jamming threshold            F-07

      .       VQCCTP ( VQC_PT ),    ! comm transmitter power            F-07


      .       VQCRSP ( VQC_PT, NC_ENV, DL_VQC ),
                                    ! thresholds of aircraft damage     F-32
                                    !  level response to nuclear
                                    !  environments 1 to NC_ENV:
                                    !  a zero entry for (VQC_PT,NC_ENV,1)
                                    !  makes environment NC_ENV not
                                    !  applicable to aircraft type VQC_PT
```

```
                                      !
                                      !    DL_VQC = 1 is loss of mission
                                      !         capability
                                      !
                                      !    DL_VQC = 2 is loss of aircraft
     .       VQCLMD ( VQC_PT),        ! distance travelled during climb    F-07
                                      !  to altitude
     .       VQCLMF ( VQC_PT),        ! fuel consumed during climb to      F-07
                                      !  altitude
     .       VQCLMA ( VQC_PT)         ! climbout altitude                  F-07


      INTEGER*4  VQCNAM ( VQC_PT, 2 )
                                      ! eight-character name for blue      F-07
                                      !  electronic warfare type


      INTEGER*2  VQCLMT ( VQC_PT),    ! time consumed during climb to      F-07
                                      !  altitude
     .           VQCJNM ( VQC_PT, 3 ),
                                      ! jammer number, corresponding to    F-07
                                      ! a jammer type.
     .           VQCJTP ( VQC_PT, 3 )
                                      ! jammer type, (integer from         F-07
                                      ! 1 to 8)


      LOGICAL*1  VQCTDS ( VQC_PT )    ! tactical data system               F-07
                                      !  capability, true or false


      COMMON /VQCHAR/ VQCSP, VQCFBR, VQCFCR, VQCFR, VQCCJT, VQCCTP,
     .                VQCRSP, VQCLMD, VQCLMF, VQCLMA, VQCNAM,
     .                VQCLMT, VQCTDS, VQCJNM, VQCJTP
```

```
*                                                                            *
* -------------------------------------------------------------------------- *
*    VQSTAT COMMON BLOCK                       04/10/86                      *
* -------------------------------------------------------------------------- *
*                                VQ status                                   *

*                                             CONTENT           INITIAL VALUE
*                                                                  OR FILE


**    PARAMETER  VQ_ID = 20


      INTEGER*2
     .           VQCNT,               ! number of VQs that have           **
                                      !  entered game

     .           VQSPTY,              ! number of electronic support     F-07
                                      !  aircraft defined

     .           VQPNTR( VQ_ID),      ! The  pointer which links the
                                      ! VQ common to the AP common

     .           VQCPID( VQ_ID),      ! the  assigned station identification
                                      ! for this electronic support
                                      ! aircraft.

     .           VQSTAT ( VQ_ID ),    ! electronic support aircraft state
                                      !  ( 0 = on deck
                                      !    1 = post-launch climbout
                                      !    2 = enroute to station
                                      !    3 = on station
                                      !    9 = Return to Base )

     .           VQSST ( VQ_ID )      ! speed status                       ---
                                      !  ( 1 = max endurance;
                                      !    2 = max range;
                                      !    3 = buster; and,
                                      !    4 = patrolling)

      COMMON /VQSTAT/ VQCNT, VQPNTR, VQSPTY, VQCPID, VQSTAT, VQSST
```

```
*                                                                          *
* ------------------------------------------------------------------------ *
*     VWCHAR COMMON BLOCK                           12/07/84               *
* ------------------------------------------------------------------------ *
*                 Blue early warning (VAW) characteristics                 *

*                                                CONTENT        INITIAL VALUE
*                                                                  OR FILE

**    PARAMETER  VWC_PT = 5,         ! blue early warning platform type      **
**                                    ! Note: this number must be less than  **
**                                    ! the number of fighter platform types **
**                                                                           **
**   .           NC_ENV = 8           ! nuclear environment ( see /ENV/ )    **

**                                                                           **
      PARAMETER  DL_VWC =  2          ! damage level                         **


      REAL*4  VWCSP ( VWC_PT, 4 ),    ! speed ( 1 = max endurance,       F-09
                                      !         2 = max range,
                                      !         3 = buster
                                      !         4 = patrol)
                                      !   meters per second

     .        VWCFBR ( VWC_PT, 4 ),   ! fuel burn rate at four           F-09
                                      !  speeds, kilograms per second

     .        VWCFCR ( VWC_PT, 4 ),   ! fuel consumption rate,           F-09
                                      !  kilograms per meter

     .        VWCFR ( VWC_PT ),       ! reserve fuel                     F-09

     .        VWCCJT ( VWC_PT ),      ! comm jamming threshold           F-09

     .        VWCCTP ( VWC_PT ),      ! comm transmitter power           F-09

     .        VWCRSP ( VWC_PT, NC_ENV, DL_VWC ),
                                      ! thresholds of aircraft damage    F-32
                                      !  level response to nuclear
                                      !  environments 1 to NC_ENV:
                                      !  a zero entry for (VWC_PT,NC_ENV,1)
                                      !  makes environment NC_ENV not
                                      !  applicable to aircraft type VWC_PT
                                      !
                                      !    DL_VWC = 1 is loss of mission
```

```
                                   !        capability
                                   !
                                   !   DL_VWC = 2 is loss of aircraft

     .        VWCLMD ( VWC_PT),    ! distance travelled during climb   F-09
                                   !  to altitude
                                   !  ( 1 = normal)

     .        VWCLMF ( VWC_PT),    ! fuel consumed during climb to     F-09
                                   ! altitude

     .        VWCLMA ( VWC_PT)     ! climbout altitude                 F-09


      INTEGER*4  VWCNAM ( VWC_PT, 2 )
                                   ! eight-character name for blue     F-09
                                   ! early warning type


      INTEGER*2  VWCSTY ( VWC_PT ),  ! sensor type                     F-09

     .           VWCADT ( VWC_PT ),  ! tracking capacity               F-09

     .           VWCLMT ( VWC_PT),   ! time consumed during climb to   F-09
                                     !  altitude

     .           VWCVC( VW_ID),      ! max number of VF that can be    F-09
                                     ! vectored to controller VW_ID

     .           VWCVF( VW_ID)       ! max number of VF that can be    F-09
                                     ! assigned to controller VW_ID


      LOGICAL*1  VWCTDS ( VWC_PT )   ! tactical data system            F-09
                                     !  capability, true or false


      COMMON /VWCHAR/ VWCSP, VWCFBR, VWCFCR, VWCFR, VWCCJT, VWCCTP,
     .                VWCRSP, VWCLMD, VWCLMF, VWCLMA, VWCNAM, VWCSTY,
     .                VWCADT, VWCLMT, VWCTDS, VWCVC, VWCVF
```

```
*                                                                          *
* ------------------------------------------------------------------------ *
*     VWSTAT COMMON BLOCK                          07/08/85                *
* ------------------------------------------------------------------------ *
*                                  VW status                               *

*                                                CONTENT        INITIAL VALUE
*                                                                  OR FILE


**    PARAMETER  VW_ID = 20


      INTEGER*2  VWCNT,                ! number of VWs that have            **
                                       !  entered game

     .           VWSPTY,               ! number of airborne early warning  F-09
                                       !  aircraft defined

     .           VWCPID( VW_ID),       ! the  assigned station identification
                                       ! for this airborne early warning
                                       ! aircraft.

     .           VWPNTR( VW_ID),       ! the pointer which links the       F-09
                                       ! VW common to the AP common

     .           VWSTAT ( VW_ID ),     ! airborne early warning aircraft state
                                       !  ( 0 = on deck
                                       !    1 = post-launch climbout
                                       !    2 = enroute to station
                                       !    3 = on station
                                       !    9 = Return to Base )

     .           VWSST ( VW_ID )       ! speed status                       ---
                                       !  ( 1 = max endurance;
                                       !    2 = max range;
                                       !    3 = buster; and,
                                       !    4 = patrolling)

      COMMON /VWSTAT/ VWCNT, VWSPTY, VWCPID, VWPNTR, VWSTAT, VWSST
```

3.6 PROGRAMMING CONVENTIONS

3.6.1 FORTRAN Conventions

FORTRANIV FEATURES, IBM FORTRAN IV (H Extended) was used. The program includes the following FORTRAN IV features not in ANS FORTRAN:

- Data types INTEGER*2 and LOGICAL*1 are used in common arrays to reduce memory requirements.

- The END parameter is used in READ s

CHAPTER 5

DETECTION AND TRACKING MODULE

The detection and tracking module models the detection and loss of detection of Red targets by Blue observers:

- o fighter aircraft;
- o early warning aircraft; and,
- o ships

The primary sensor modeled is radar, with both active detections and passive identifications of jamming determined. Visual detections are modeled for fighter aircraft only. Although passive sensors, such as Electronic Support Measures (ESM), are part of the defense system; they are not currently modeled. Future sensor systems such as Infra-Red Search and Track (IRST) are not currently modeled in NADS.

The operations of fleet air defense are initiated by the detection process. It it through this process that the necessary data to make decisions, issue force orders, and engage the threat is obtained. The quality of the air defense simulation depends on the fidelity of the simulation of detection and loss of detection.

The detection calculations are complicated because actions taken by the attack and the defense impact the detection process. Whenever an action is taken, the detections influenced by that action must be recalculated. As a minimum, detections for the maneuvering unit combined with all units of the other side must be computed. In a large scenario, the number of combinations can be large. In a counter-measures enviroment, detections of units not involved in the actions can also be affected. As a consequence, the detection and tracking module for NADS is the most frequently used and the most significant consumer of computer time.

To reduce computer run time for the model, several assumptions are made:

- o Aircraft in holding, flying a fixed pattern based on a position, are considered to have a detection pattern which can be approximated by a circle.

- o Ships are stationary throughout any simulation run.

- o Detection will not change during large turns by Red and Blue aircraft; the calculations are made at the initiation of the turn and then again at the end of the turn.

To save on run time for the model, several complicated calculations have been moved from the detection and tracking module to both the Red Scenario Module and the Blue aircraft modules:

- o Horizon crossing; and,

- o Sector crossing for fighter aircraft.

These calculations are performed once and the results stored. When the calculations were performed within the detection and tracking module, the same computation was frequently duplicated.

To save on computer run time for the model, times when detections calculated will no longer be valid are computed in the Red Scenario Module, in the Fighter Module, and in the Early Warning Module. Essentially, these times are the times for the next maneuver for the unit when the detection will be recalculated. These times are saved and used in the detection routines to avoid calculations which will later be invalidated.

5.1 BASIC DETECTION LOGIC

The detection module uses the radar equation to calculate the times when detection and loss of detection in a clear environment is possible. If detection is not possible before the time that detection is not valid, no further calculations are made. A different form for the radar equation is used to compute detection time and loss of detection time in the current jamming enviroment. The detection module uses a fixed range to calculate times when detection and loss of detection are possible for visual detection. Detection times are delayed to account for operational factors.

5.1.1 Radar Detection

Using Subroutine XHOR, the rise and set of the target for the observers radio horizon is computed; these times bound the detecton. If the sensor has a sector field-of-view, the crossing times for the edges of the sector which further bound the detection are calculated using Subroutine SECTOR.

The detection module uses the radar equation to calculate the times when detection and loss of detection are possible. The nominal detection range (Range Beta) of each radar type is among the input data. It represents the radar's clear environment capability against a one square meter target when there is no limitation of operator skill, exhaustion, or division of attention by other targets. Range Beta is scaled by target radar cross section to calculate a detection range in the clear. This range is bounded by the instrumented range for the radar. Detection in the clear is feasible when the target crosses this range.

5.1.2 Radar Jamming Detection

Jammers are assumed to be wideband noise jammers. Standoff jammers can be positioned by the scenario or jammers can be placed on attacking aircraft. In any case, the affect of the jammer is dependent on the changing geometry between:

1. the jammer;
2. the target; and,
3. the observer.

as well as the characteristics of the jammer and the observer's radar.

The directivity of the jammer's antenna is represented in the data for effective radiated power density in the victim radar's bandwidth. Jammer antenna patterns are usually broad enough to cover all the intended vicitms. NADS assumes that is the case, and does not require that the scenario include jammer aiming data.

The times when jammers transition from mainlobe jammers to sidelobe jammers are calculated. The radar is assumed to be looking at the target. A simplified model of the observer's radar beam pattern is assumed. Either a two-dimensional pattern or a three-dimensional pattern is possible. For the two-dimensional pattern the main beam is fan shaped with a constant width in elevation. For the three-dimensional pattern the main beam is conical in shape. The antenna pattern consists of this single main lobe with a constant sidelobe in all other directions. The width of the main lobe and the gain of the sidelobe relative to the main lobe are part of the input hardware characteristics.

The burnthrough range is computed as

$$R_b^4 = R_a^2 \times A \times \left(\frac{S_1 \times P_1}{D_1^2} + \frac{S_2 \times P_2}{D_2^2} + \ldots + \frac{S_n \times P_n}{D_n^2} \right)^{-1}$$

where

R_b is the burnthrough range,

R_a is Range Alpha, burnthrough range on a one square meter target self-screen jammed by one watt per megahertz,

S_n is the radar's sidelobe ratio (arithmetic, not dB) if the jammer is in the sidelobe and one otherwise,

A is the radar cross section of the target,

P_n is the effective radiated power density of jammer number n, in watts per megahertz, and

D_n is the distance between the radar and jammer n.

The number of jammers is dependent on the number of jammers assigned to this sensor in the scenario which are above the horizon and in the sector of the radar. There is a discontinuity whenever:

- a jammer transitions from the sidelobe to the main lobe;
- a jammer transitions from the main lobe to the sidelobe;
- a jammer enters or leaves the field of view of the radar;
- a jammer rises or sets on the observer's radio horizon; or,
- the allocation of coherent side lobe cancellers changes.

The method used in NADS is to step from discontinuity to discontinuity in the interval from time of detection in the clear to the time of lost detection in the clear looking for burnthrough transitions on the target.

All Red units in NADS must move with a speed greater than one meter per second. A closed form solution to the above equation is not feasible when the jammers are moving; the burnthrough time must be known before the correct distance values can be obtained to compute the burnthrough range. Consequently, an interative solution, Muller's Method for finding the zero's for an arbitrary function, is used to solve the equation. While the procedure converges quadratically, it can add substantially to the run time.

5.1.3 Visual Detection

The detection module uses a fixed range to calculate times when detection and loss of detection are possible for visual detection. These times are delayed in the same fashion as those for radar detections.

5.1.4 Detection Delays

Time detected is the time an operator would actually make a detection and includes delays for classifying a target. This delay is a function of the track storage contents. An empirical relationship is used to fit radar equation results to observed data by adding a delay to the time of detection. For sensors without a full 360 degree field of view, an additional random delay is imposed to account for the orientation of the aircraft.

Two decision delays for target identification are imposed:

- o The Red unit detected is not assigned to the Blue observer.
- o The Red unit detected has been assigned to this Blue unit so that the Blue unit would be looking for the Red unit.

Since each unit has a limit on the number of targets which can be tracked, a contact is delayed until there is room for the target to be tracked. Logic to preempt contacts for assigned targets is furnished.

5.2 FORTRAN SUBPROGRAMS REQUIRED

The following is a list of the FORTRAN subprograms required to operate the Detection and Tracking module. Figure 5-1, FORTRAN Subroutine Usage, shows the routines used and the calling routines.

	BURN	CUBIC	DETDLY	DETECT	DMULLER	MKMSG	MNLJ3D	MNLJAM	MOVEL	QUADRATIC	QUARTIC	SECTOR	SSJAM	TRACK	XHOR
BURN															
CUBIC															
DETDLY															
DETECT	•				•		•	•	•	•					
DMULLER	•														
MKMSG															
MNLJ3D											•				
MNLJAM										•					
MOVEL															
QUADRATIC															
QUARTC		•								•					
SECTOR															
SSJAM						•			•	•					
TRACK						•									
XHOR										•	•				
HLPRTN			•	•									•	•	
RANODE												•			•
RMNODE												•			•
APLNCH												•			•
VFCALC												•			•
VQCALC												•			•
VWCALC												•			•
VWPTRL												•			•

Figure 5-1 Detect and Track FORTRAN Usage

Function	
BURN	Function which computes the difference between the burnthrough range and the range from observer to target at some input time.
CUBIC	Solves for the roots of the cubic $X^3 + PX^2 + QX + R = 0.$
DETDLY	Computes the detection delay time based on input parameters; business of the sensor; target status; and, holding status of a fighter.
DETECT	Determines the next times of gain and loss of detection of a Red target by a Blue observer. Determines if the target is currently detectable. Determines the time interval when a jammer will be resolved.
DMULLER	Uses Muller's method to find any desired number of zeros, real or complex, of an arbitrary function. (double precision version for G-FLOATING option)
MKMSG	Message composition utility.
MNLJ3D	Calculates the times a jammer enters and leaves the main beam of the radar, three-dimensional, when the radar is pointed at a target.
MNLJAM	Calculates the times a jammer enters and leaves the main beam of the radar, two-dimensional, when the radar is pointed at a target.
MOVEL	Updates the list of the moving BLUE aircraft and/or RED units. The amount of fuel remaining in BLUE fighters is also updated to the current clock time.
QUADRATIC	Solves for the real roots of the quadratic $AX^2 + BX + C = 0.$
QUARTC	Solves for the real roots of the quartic $AX^4 + BX^3 + CX^2 + DX + E = 0.$
SECTOR	Computes sector entry and exit times for fighter aircraft sensors.

SSJAM — Reports resolution of jamming and cessation of jamming for all Blue units. Computes the time when fighters will perceive that a jammer is close. This perception is currently based on jammer elevation angle or bearing rate.

XHOR — Computes rise time above the radio horizon and the set time below the radio horizon for a Red unit, Blue unit pair.

Function XYDIST — Computes the two dimensional distances between two points.

5.2.1 Subroutine CUBIC

Subroutine CUBIC is a utility routine used to solve a cubic equation. A cubic equation,

$$y^3 + py^2 + qy + r = 0$$

may be reduced to the form,

$$x^3 + ax + b = 0 \ ,$$

by substituting for y the value,

$$x - \frac{p}{3} \ , \quad \text{where}$$

$$a = \frac{1}{3} \left(3q - p^2 \right) , \text{ and}$$

$$b = \frac{1}{27} \left(2p^3 - 9pq + 27r \right).$$

For solution, let,

$$A = \text{cube root}\left(- \frac{b}{2} + \text{square root}\left(\frac{b^2}{4} + \frac{a^3}{27} \right)\right),$$

$$B = \text{cube root}\left(- \frac{b}{2} - \text{square root}\left(\frac{b^2}{4} + \frac{a^3}{27} \right)\right),$$

then the values of x will be given by,

$$x = A + B ,$$

$$x = - \frac{A + B}{2} + \frac{A - B}{2} \text{ square root}(- 3) , \text{ and}$$

$$x = - \frac{A + B}{2} - \frac{A - B}{2} \text{ square root}(- 3) .$$

If p, q, r are real, then

If $\frac{b^2}{4} + \frac{a^3}{27} > 0$, there will be one real root and two conjugate imaginary roots.

If $\frac{b^2}{4} + \frac{a^3}{27} = 0$, there will be three real roots of which at least two are equal.

If $\frac{b^2}{4} + \frac{a^3}{27} < 0$, there will be three real and unequal roots.

In the last case a trigonometric solution is used by computing the value of the angle theta in the expression,

$$\text{theta} = \arccos\left(- \frac{b}{2} / \text{square root}\left(- \frac{a^3}{27}\right)\right) ,$$

then x will have the following values:

$$x = 2 \text{ square root } \left(- \frac{a}{3} \right) \cos \left(\frac{\text{theta}}{3} \right) ,$$

$$x = 2 \text{ square root } \left(- \frac{a}{3} \right) \cos \left(\frac{\text{theta}}{3} + 120 \right) , \text{ and}$$

$$x = 2 \text{ square root } \left(- \frac{a}{3} \right) \cos \left(\frac{\text{theta}}{3} + 240 \right) .$$

5.2.2 Subroutine DETDLY

Subroutine DETDLY is the routine used to modify the detection performance based on the radar equation calculated in Subroutine DETECT.

The number of potentially detectable targets for this observer is maintained by the GPSS and is an input to the routine. For a manual system, you could think of the number of potentially detectable targets as the number of points where the radar display is glowing. The average time delay for targets is described by the formula:

$$\text{lambda} = a + bN^2 , \text{ where}$$

a and b are constants associated with the sensor system, and N is the number of potential targets.

Since a sensor system will be looking for an assigned target, a different average delay is appropriate. In DETDLY, a constant average delay which is a function of the sensor system is applied:

$$\text{lambda} = c .$$

If the sensor is installed on a fighter in holding, detection is calculated with the assumption that the sensor has a 360 degree field of view. In DETDLY, a random heading is specified by drawing a uniform pseudo-random number and the time until the sector of the sensor includes the target is calculated.

The actual delay is determined by drawing a pseudo-random number from the exponential distribution with parameter lambda. The time to turn until the sector includes the target is added to the actual delay for fighters in holding.

5.2.3 Subroutine DETECT

Subroutine DETECT is the routine used to calculate the times of detection and lost detection. A visual detection based on a fixed range is computed for aircraft detecting aircraft. However, the main purpose of the routine is to solve the radar equation. The subroutine is divided into seven sections. The logical relationship between these sections is shown in Figure 5-2, Flow Diagram for Subroutine DETECT.

The algorithm used in DETECT assumes that the velocity of the target, the observer and any jammers influencing the detection are constant. Whenever a change of velocity happens, the times for detection and lost detection must be recomputed with the new velocity.

Detection calculation is the single most significant consumer of computer time within NADS. The detection process is limited by a number of physical phenomena:

Horizon crossing;

Sector crossing;

Detection range in the clear of the radar;

Jamming intensity; and,

Detection range when jammed of the radar.

The process is also limited by the future maneuvers. In NADS, knowledge of the future is used to limit the region of interest. The region is a time interval which is initially bounded on the left by the current time and on the right by future maneuvers. The subroutine is designed to terminate as soon as a condition is discovered that describes the detection performance in that interval.

The first section of DETECT is a preparation section in which:

control variables are initialized;

the target and observer positions are updated to the current time; and,

positions and velocities are converted to an observer-centered coordinate system.

The logical flow for this section is shown in Figure 5-3, Flow Diagram for Subroutine DETECT (Section I).

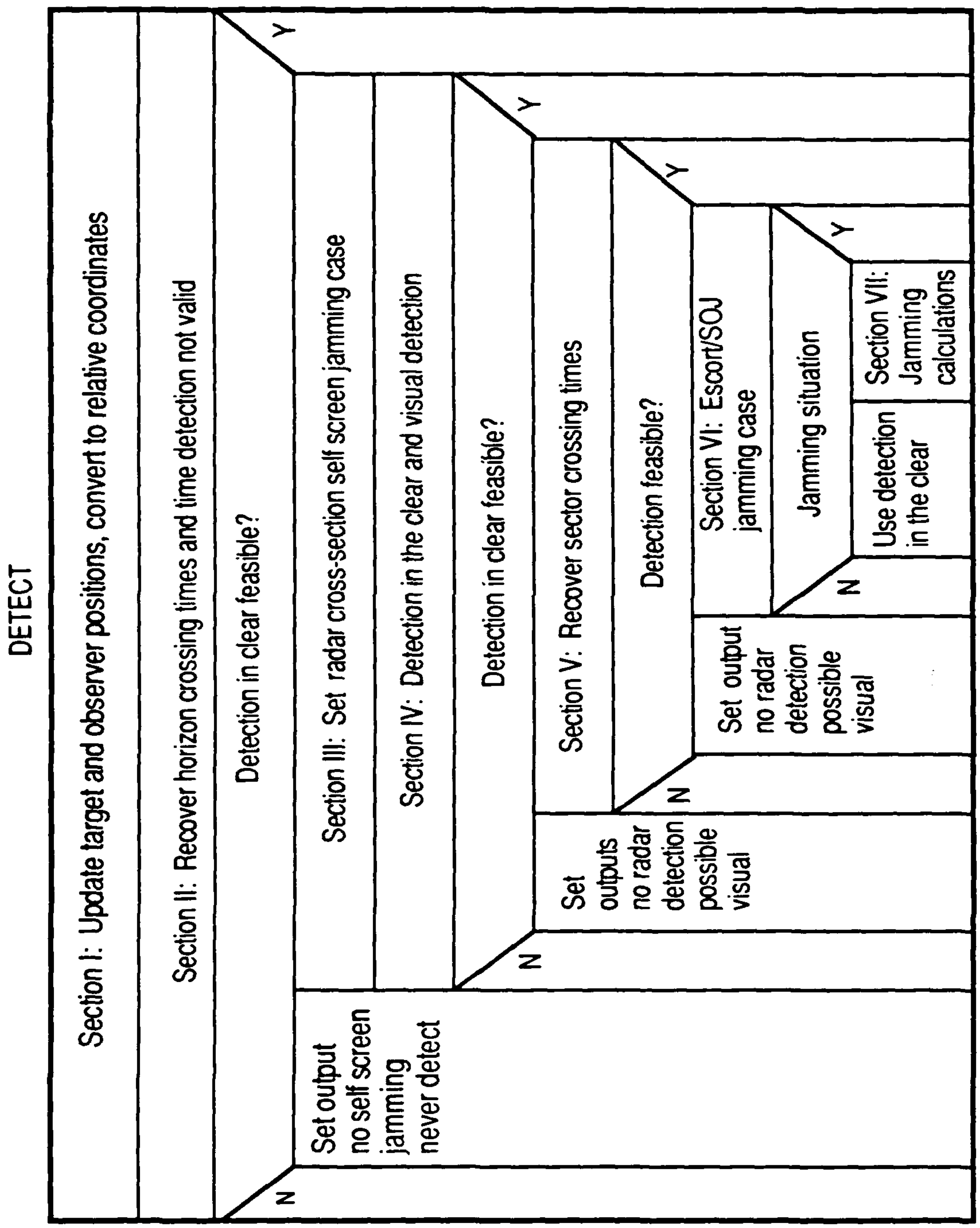


Figure 5-2 Flow Diagram for Subroutine Detect

The second section of DETECT recovers the horizon crossing times which have been calculated by Subroutine XHOR in the Red Threat Module, the Fighter Module and the Early Warning Aircraft Module. The initial region of interest is also established in this section of the subroutine. The logical flow for this section is shown in Figure 5-4, Flow Diagram for Subroutine DETECT (Section II).

If the target never rises above the radio horizon of the observer during the bounded interval, detection will never happen until something changes. In this case, the calculations are complete.

The third section of DETECT recovers the radar cross-section for the target from the appropriate characteristics data base and decides if this is a self-screen jamming case; i.e., the target is jamming the observer. With self-screen jamming, all the data arrays used to compute detection in a jamming enviroment are initialized. Additionally, the interval is computed during which the target is the strongest jammer in the main lobe of the radar with the radar looking at the target.

This special jamming case is considered at this point because the next bound on the detection interval will be radar detection in the clear which is limited by the size of the target and the characteristics of the radar. The jamming phenomena is not constrained in the same way. The logical flow for this section is shown in Figure 5-5, Flow Diagram for Subroutine DETECT (Section III).

The fourth section of DETECT calculates times for visual detection and radar detection in the clear. The detection range for the radar is limited by the scaled range beta and the instrumented range. The closest point of approach (CPA) for the target to the observer is computed and compared with the detection range for the radar. With a CPA less than the detection range, the times that the target is at the detection range is calculated and used to be the time of detection and lost detection. The logical flow for this section is shown in Figure 5-6, Flow Diagram for Subroutine DETECT (Section IV).

The interval of interest is further bounded by detection in the clear. If there is no interval left, the calculations are complete.

The fifth section of DETECT recovers the sector crossing times for fighter sensors which were calculated by Subroutine SECTOR in the Red Threat Module and the Fighter Module. The logical flow for this section is shown in Figure 5-7, Flow Diagram for Subroutine DETECT (Section V).

The interval of interest is further bounded by times entering and leaving the field of view of the sensor. If there is no interval left, the calculations are complete.

In section six of DETECT, all the data arrays used to compute detection in a jamming environment are initialized for the escort jammer or standoff jammer cases. The interval of interest is further bounded by the time when detection will be invalidated for each effective jammer in the scenario. The logical flow

DETECT - SECTION I

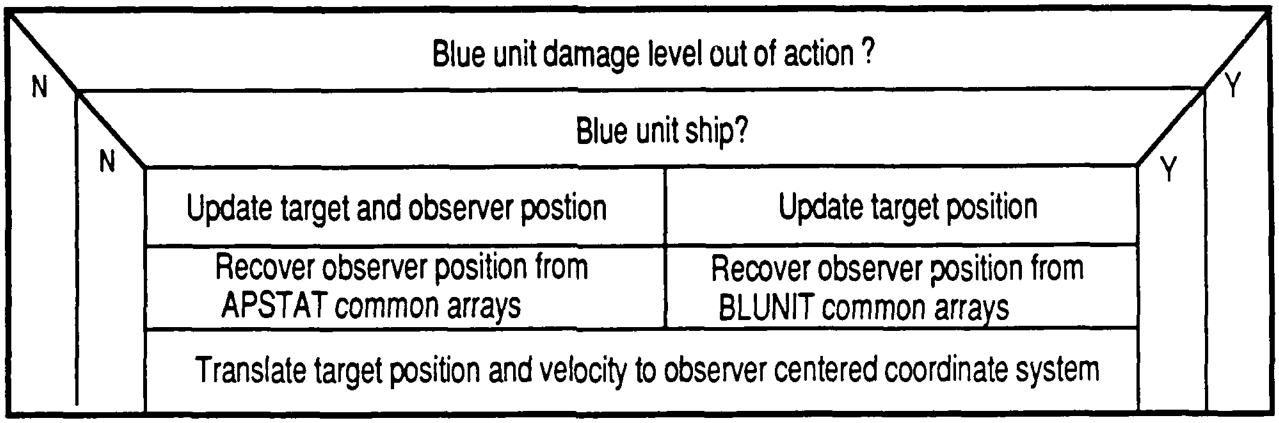


Figure 5-3 Flow Diagram for Subroutine DETECT (Section I)

DETECT - SECTION II

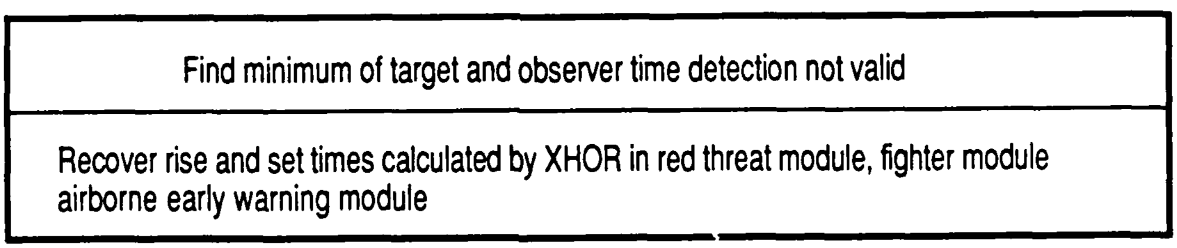


Figure 5-4 Flow Diagram for Subroutine DETECT (Section II)

DETECT - SECTION III

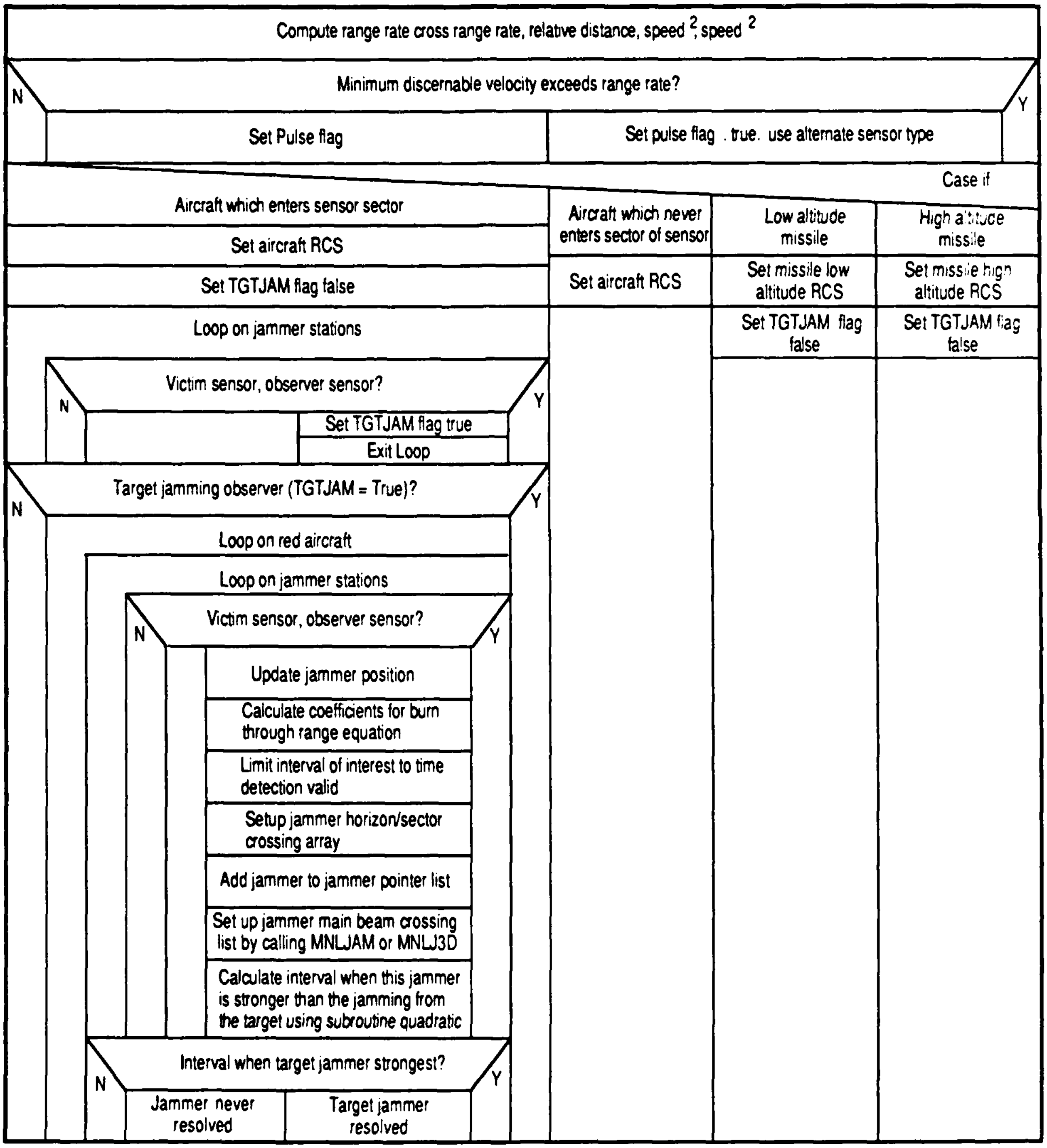


Figure 5-5 Flow Diagram for Subroutine DETECT (SECTION III)

DETECT SECTION IV

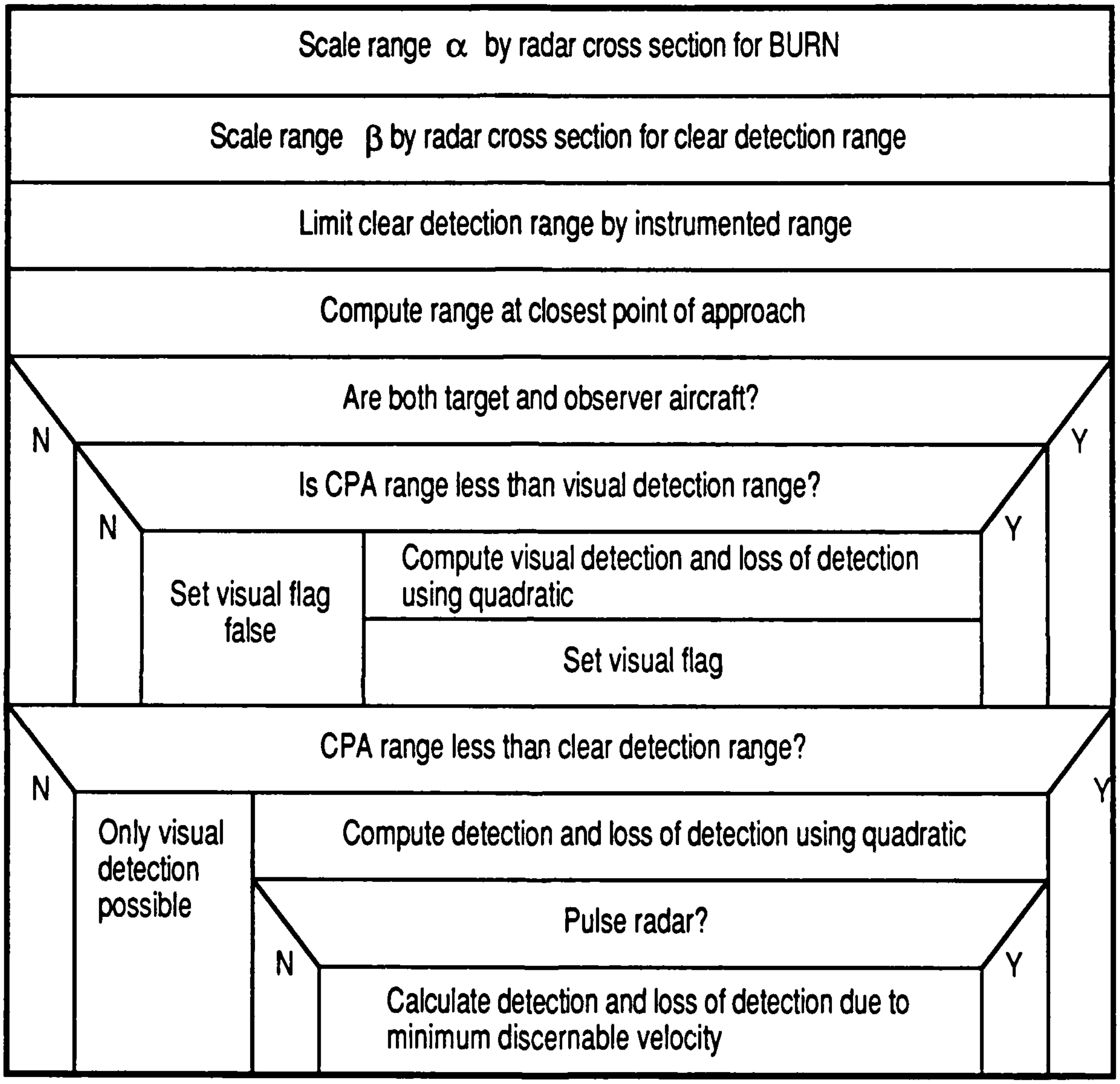


Figure 5-6 Flow Diagram for Subroutine DETECT (Section IV)

DETECT - SECTION V

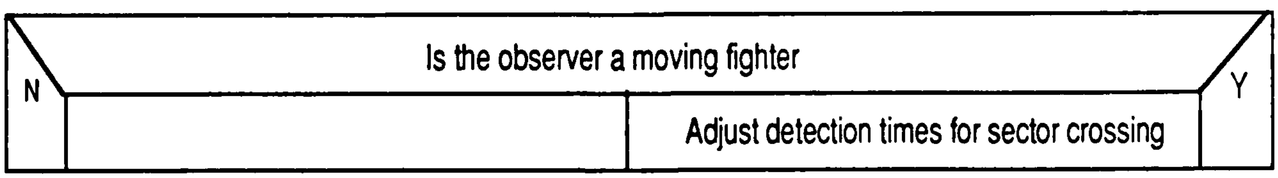


Figure 5- 7 Flow Diagram for Subroutine DETECT (Section V)

for this section is shown in Figure 5-8, Flow Diagram for Subroutine DETECT (Section VI).

The seventh section of DETECT computes the times of detection and lost detection in jamming by computing burn through times by finding when the range-to-go to burnthrough, Function BURN, is zero. Only the interval of interest is considered.

Burnthrough is not necessarily a continuous function over the interval of interest. A point of discontinuity is caused whenever

a jammer rises above the observers horizon;

a jammer sets below the observers horizon;

a jammer enters the field of view of the radar;

a jammer leaves the field of view of the radar;

a jammer enters the main beam of the radar;

a jammer leaves the main beam of the radar; or,

the allocation of coherent sidelobe cancellers changes.

The procedure used in DETECT is: step from discontinuity to discontinuity across the interval of interest, checking each point of discontinuity with Function BURN and the subinterval to the next point of discontinuity with Subroutine MULLER. Subroutine MULLER also calls Function BURN. The logical flow for this section is shown in Figure 5-9, Flow Diagram for Subroutine DETECT (Section VII).

5.2.4 Subroutine MNLJ3D

Subroutine MNLJ3D caluclates the times a jammer will enter and leave the main lobe of an observers's radar when the radar is pointed at a target. The main beam of the radar is modeled as a cone with vertex at the radar. An observer centered coordinate system is used. The relative time when the dot product of the position vector for the jammer with the position vector of the target is equal to the cosine of the half angle of the main beam is computed using Subroutine QUARTC. The complete logical flow is shown in Figure 5-10, Flow Diagram for Subroutine MNLJ3D.

DETECT - SECTION VI

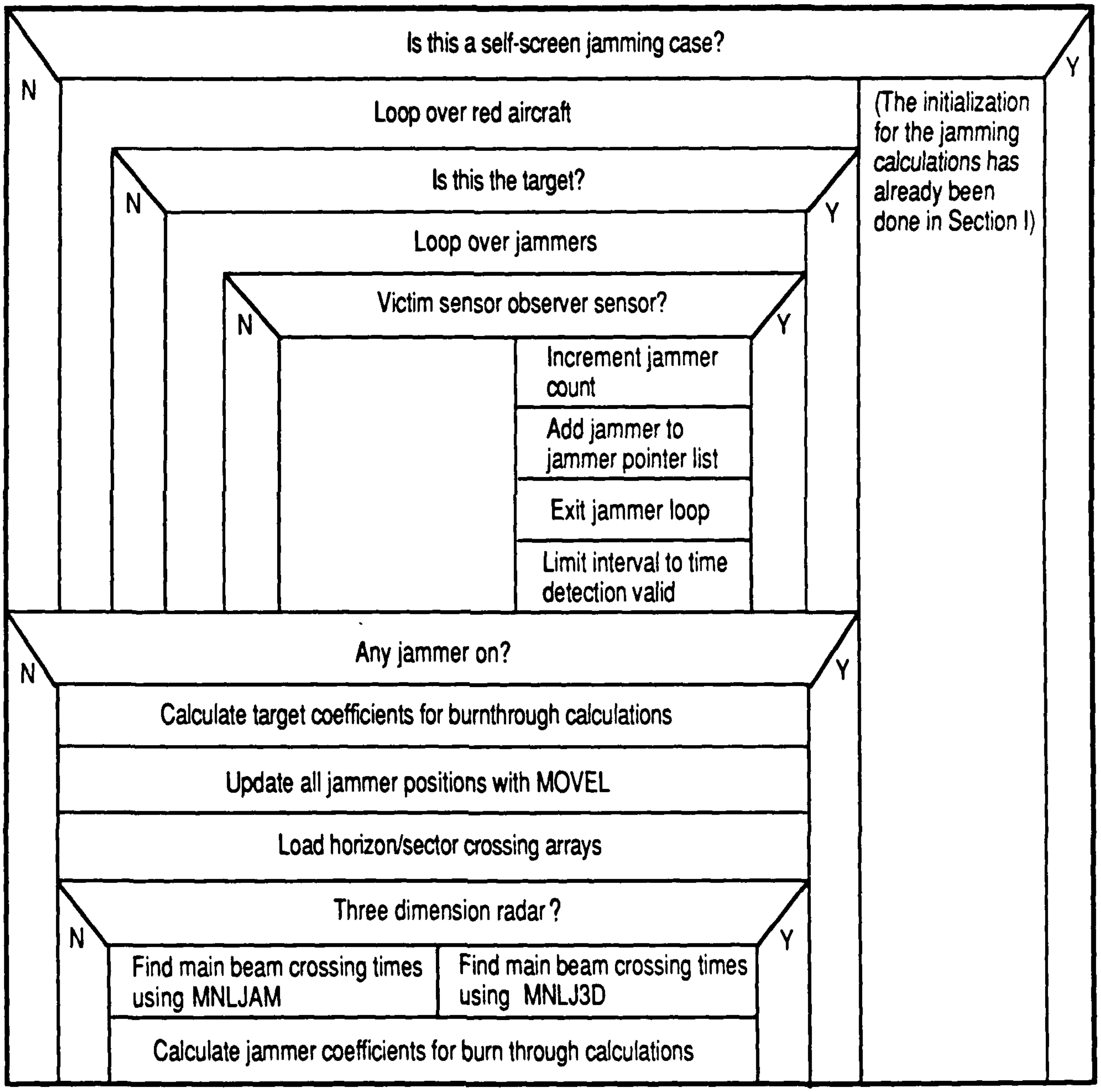


Figure 5-8 Flow Diagram for Subroutine DETECT (Section VI)

DETECT - SECTION VII

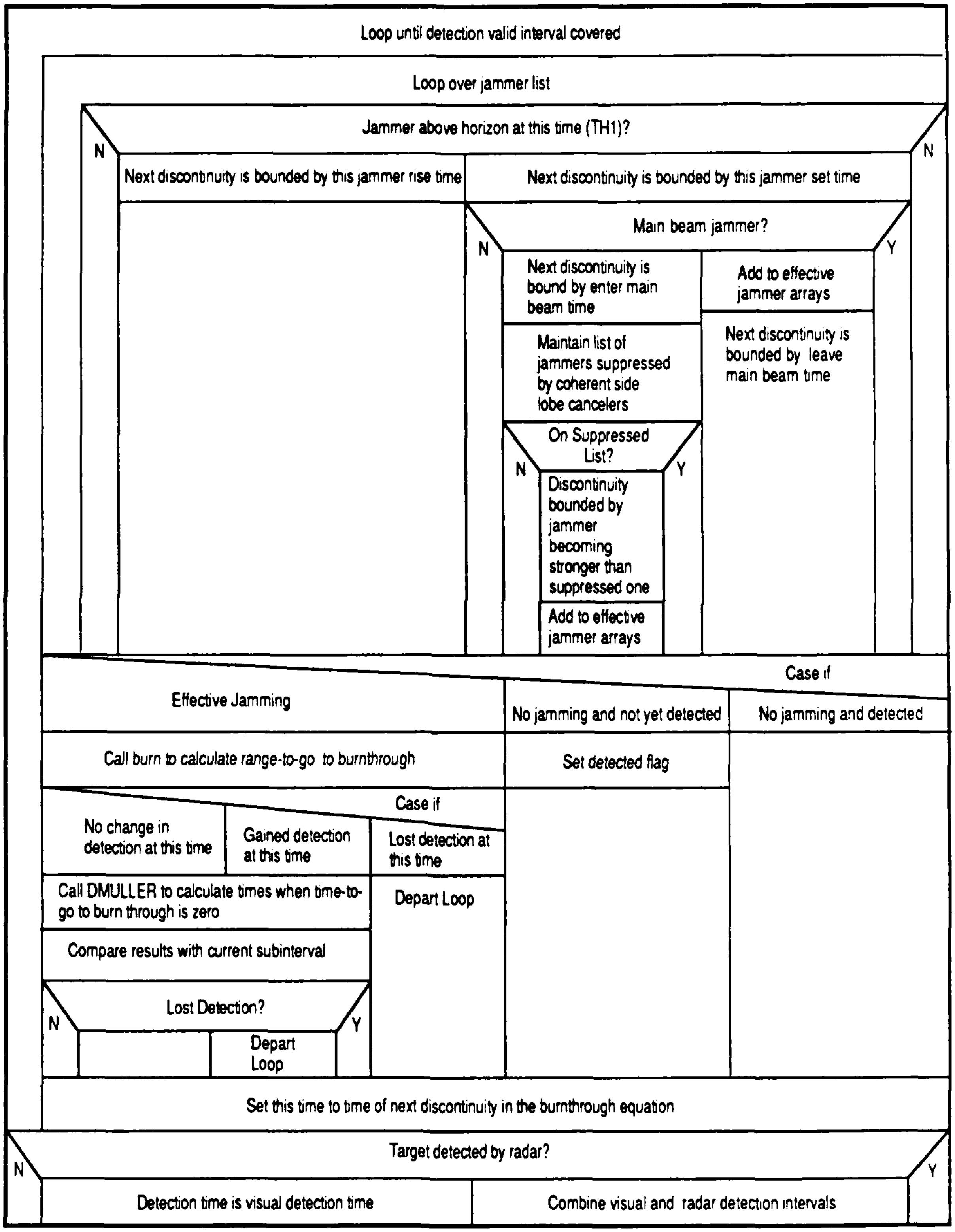


Figure 5-9 Flow Diagram for Subroutine DETECT (Section VII)

MNLJ3D

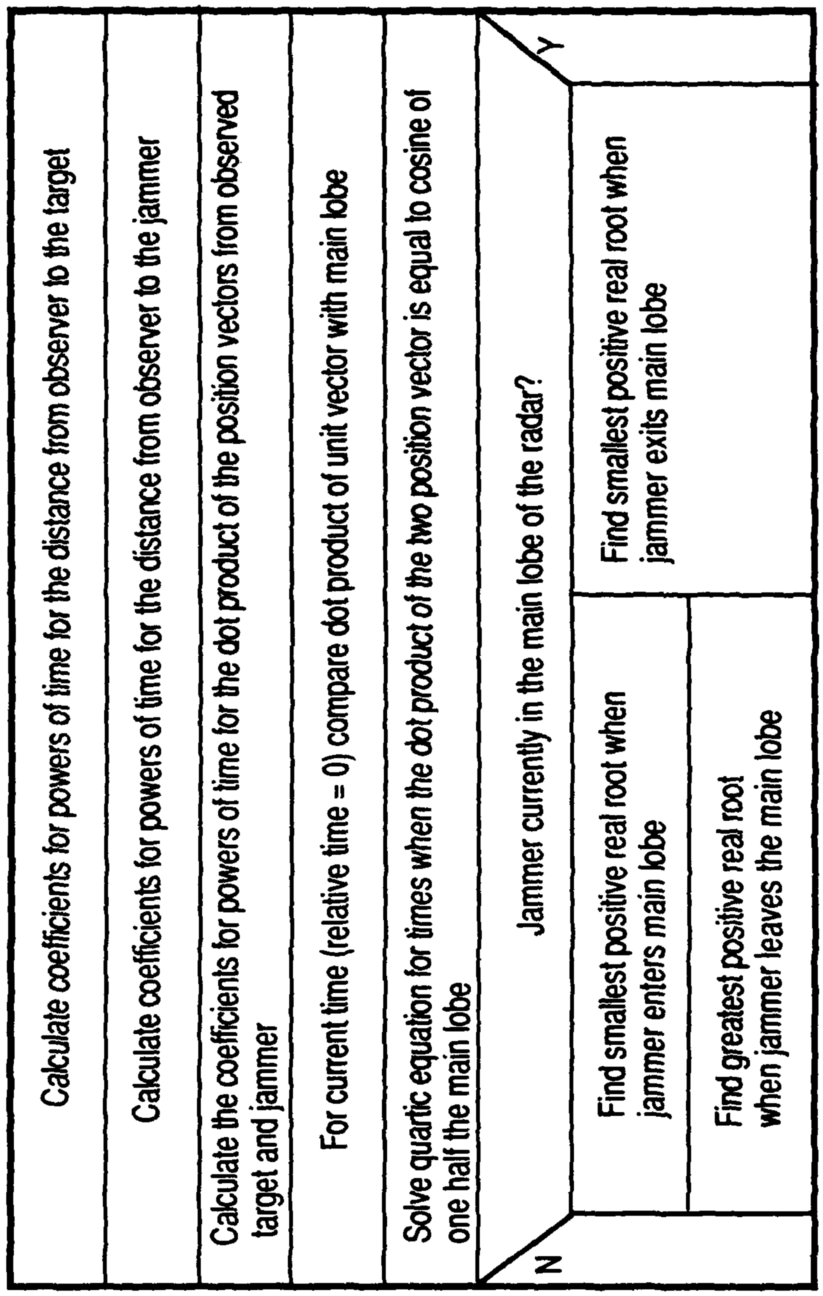


Figure 5-10 Flow Diagram For Subroutine MNL3D

5.2.5 Subroutine MNLJAM

Subroutine MNLJAM caluclates the times a jammer will enter and leave the main lobe of an observers's radar when the radar is pointed at a target. The main lobe is characterized by a constant angular width in the horizonal. An observer centered coordinate system is used. The X-axis is rotated to the edge of the main lobe as the mainlobe tracks the target. The relative time when the position vector for the jammer in that coordinate system crosses the edge is computed using Subroutine QUADRATIC. The complete logical flow is shown in Figure 5-11, Flow Diagram for Subroutine MNLJAM.

5.2.6 Subroutine QUADRATIC

Subroutine QUADRATIC is a utility routine used to solve for the real roots of a quadratic equation of the form:

$$aX^2 + bX + c = 0 .$$

If $b^2 < 4ac$ then there are no real roots.

If $b^2 = 4ac$ then there are two identical real roots equal to $-\dfrac{b}{2a}$.

If $b^2 > 4ac$ then there are two distinct real roots:

$$\frac{-b + \text{square root}(b^2 - 4ac)}{2a} ; \text{ and } \frac{-b - \text{square root}(b^2 - 4ac)}{2a} .$$

5.2.7 Subroutine QUARTC

Subroutine QUARTC is a utility routine used to solve a quartic equation. A quartic equation

$$X^4 + aX^3 + bX^2 + cX + d = 0$$

has the resolvent cubic equation

$$Y^3 - bY^2 + (ac - 4d)Y - a^2d + 4bd - c^2 = 0 .$$

MNLJAM

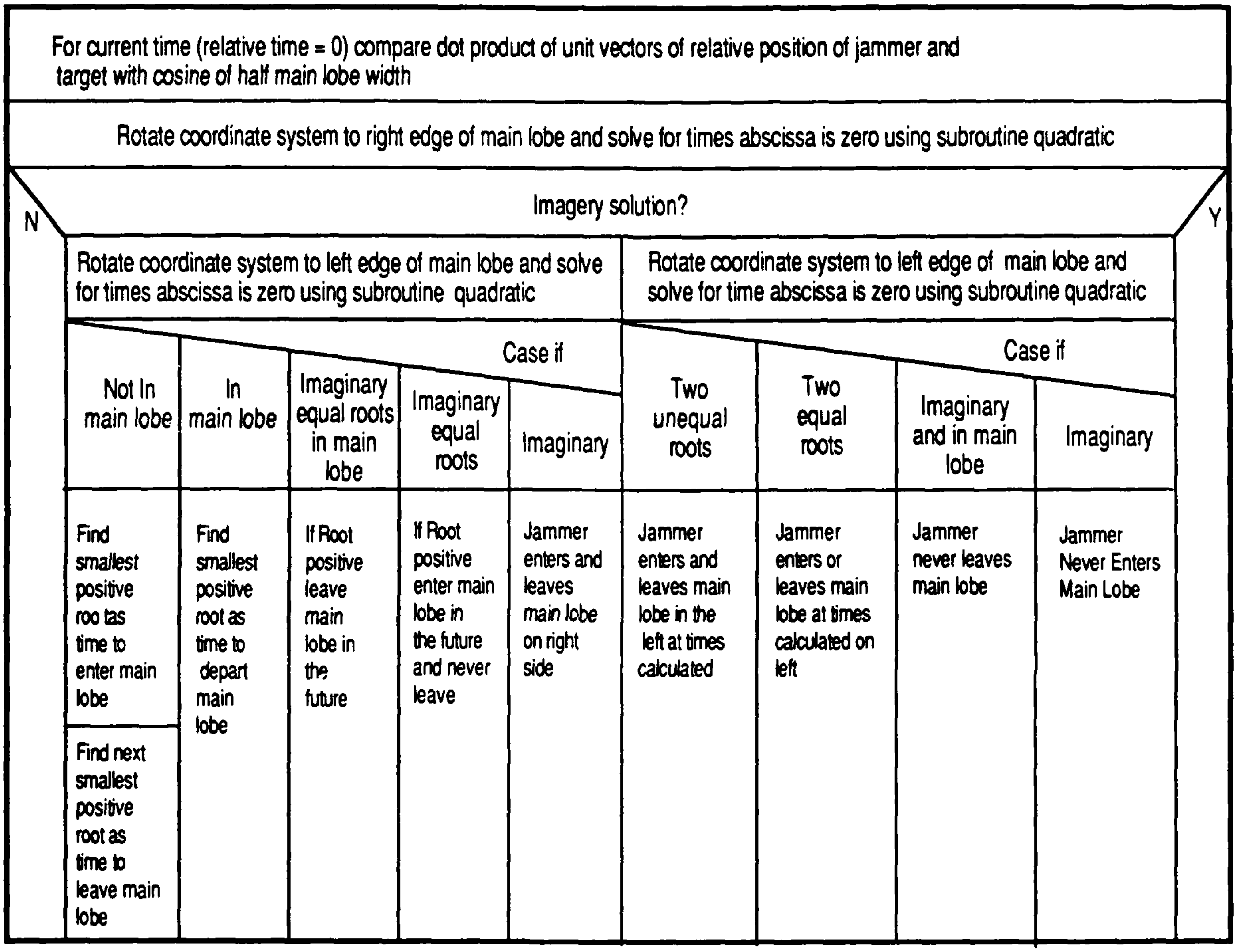


Figure 5-11 Flow Diagram for Subroutine MNLJAM

Let y be any root of this equation, and

$$R = \text{square root} \left(\frac{a^2}{4} - b + y \right).$$

If R is not equal to zero, then let

$$D = \text{square root} \left(\frac{3a^2}{4} - R^2 - 2b + \frac{4ab - 8c - a^3}{4R} \right),$$

$$E = \text{square root} \left(\frac{3a^2}{4} - R^2 - 2b - \frac{4ab - 8c - a^3}{4R} \right),$$

else if R is equal to zero, then

$$D = \text{square root} \left(\frac{3a^2}{4} - 2b + 2 \text{ square root}(y^2 - 4d) \right),$$

$$E = \text{square root} \left(\frac{3a^2}{4} - 2b - 2 \text{ square root}(y^2 - 4d) \right).$$

The four roots of the original equation are given by

$$x = -\frac{a}{4} + \frac{R}{2} + \frac{D}{2},$$

$$x = -\frac{a}{4} + \frac{R}{2} - \frac{D}{2},$$

$$x = -\frac{a}{4} - \frac{R}{2} + \frac{E}{2}, \text{ and}$$

$$x = -\frac{a}{4} - \frac{R}{2} - \frac{E}{2}.$$

5.2.8 Subroutine SECTOR

Subroutine SECTOR calculates the time a target will cross the edges of a sensor which has a sector of a circle for a field of view. The algorithim uses the dot product of two vectors to determine whether the target is currently within the sector. The special cases of the target entering or exiting the sector at the observer are checked. The target may cross the apex of the sector without entering or leaving depending on the crossing geometry and the width of the sector. The general case is solved by rotating the coordinate system to the edges of the sector and calculating the time of the position vector intercept with the X-axis in the new coordinate system. The complete logical flow is shown in Figure 5-12, Flow Diagram for Subroutine SECTOR.

5.2.9 Subroutine SSJAM

Subroutine SSJAM uses two methods to decide that a jammer is close to a fighter:

Bearing rate; and,

Elevation angle.

Both methods use Subroutine Quadratic in a fighter centered coordinate system.

SECTOR

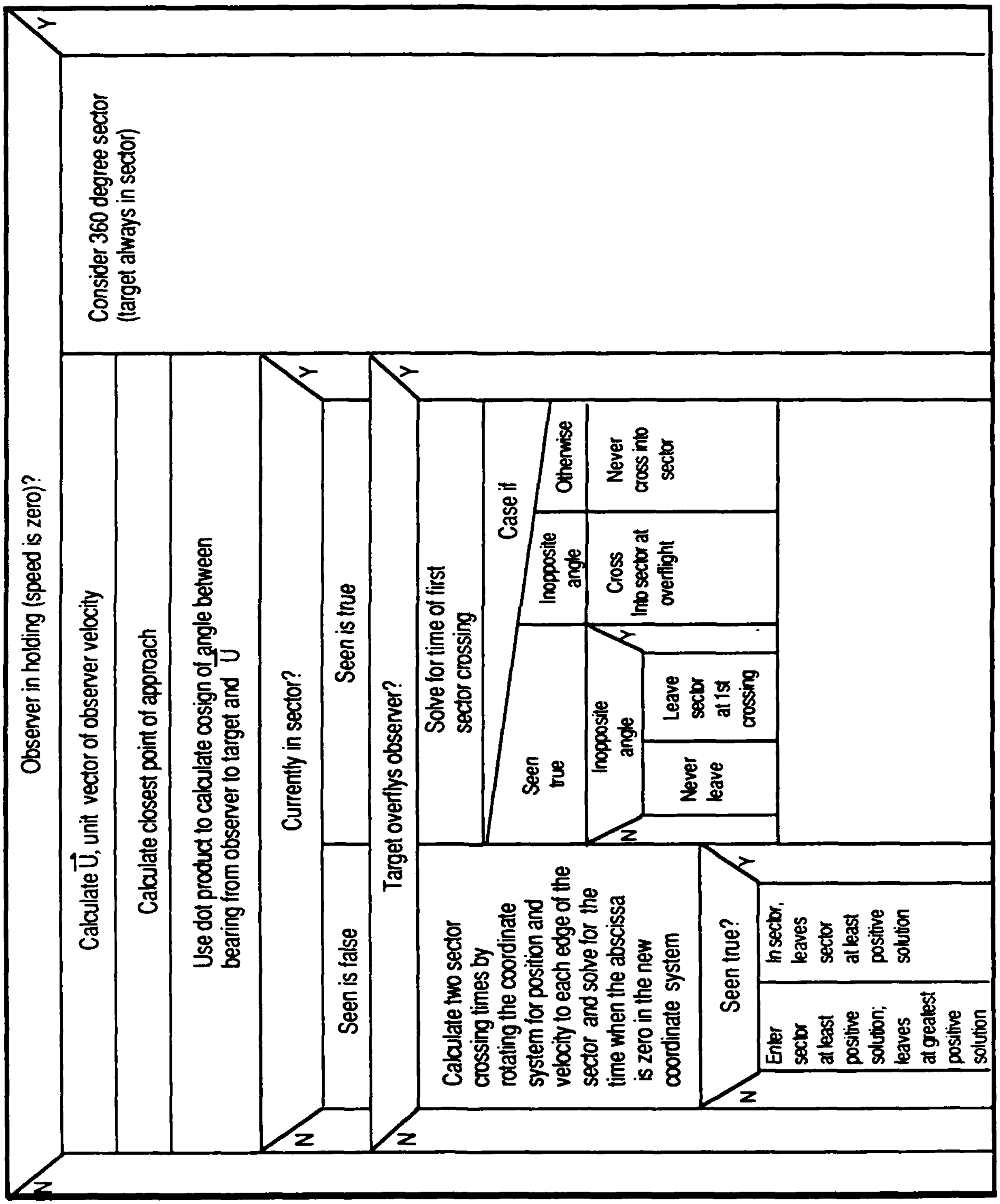


Figure 5-12 Flow Diagram for Sector

The equation:

$$A\,t^2 + B\,t + C = 0 \text{ , where}$$

A = relative speed squared ,

B = 2 x range rate, and

$$C = \text{distance}^2 - \frac{\text{absolute value(cross-range rate)}}{\text{tangent(bearing rate)}} \quad .$$

is solved when the fighter is in holding.

To determine the times when the target can be distinguished as either up or down in elevation angle, the range to the target when the required bearing is reached is determined from the law of sines for a triangle with sides of:

Radius of the earth + height of the fighter

Radius of the earth + height of the jammer

Unknown range from fighter to jammer

The elevation angle from the fighter to the jammer is the third known element of the triangle.
The following equation is then solved for time using the range.

$$A\,t^2 + B\,t + C = 0 \text{ , where}$$

A = relative speed squared ,

B = 2 x range rate, and

$$C = \text{distance}^2 - \text{Range}^2 \; .$$

If either the fighter or the jammer are climbing or diving, a simple interation scheme is used to calculate the desired times.

5.2.10 Subroutine XHOR

Subroutine XHOR uses the earth radius method of Scheleng, Burrows and Ferrell to solve for the rise and set times for one unit at the radio horizon of another with the assumption that both observer and target are at constant altitude. An additional complication happens when either unit is climbing or diving. After checking that the calculation is applicable, a quartic equation is solved for the times. The complete logical flow is shown in Figure 5-13, Flow Diagram for Subroutine XHOR.

XHOR

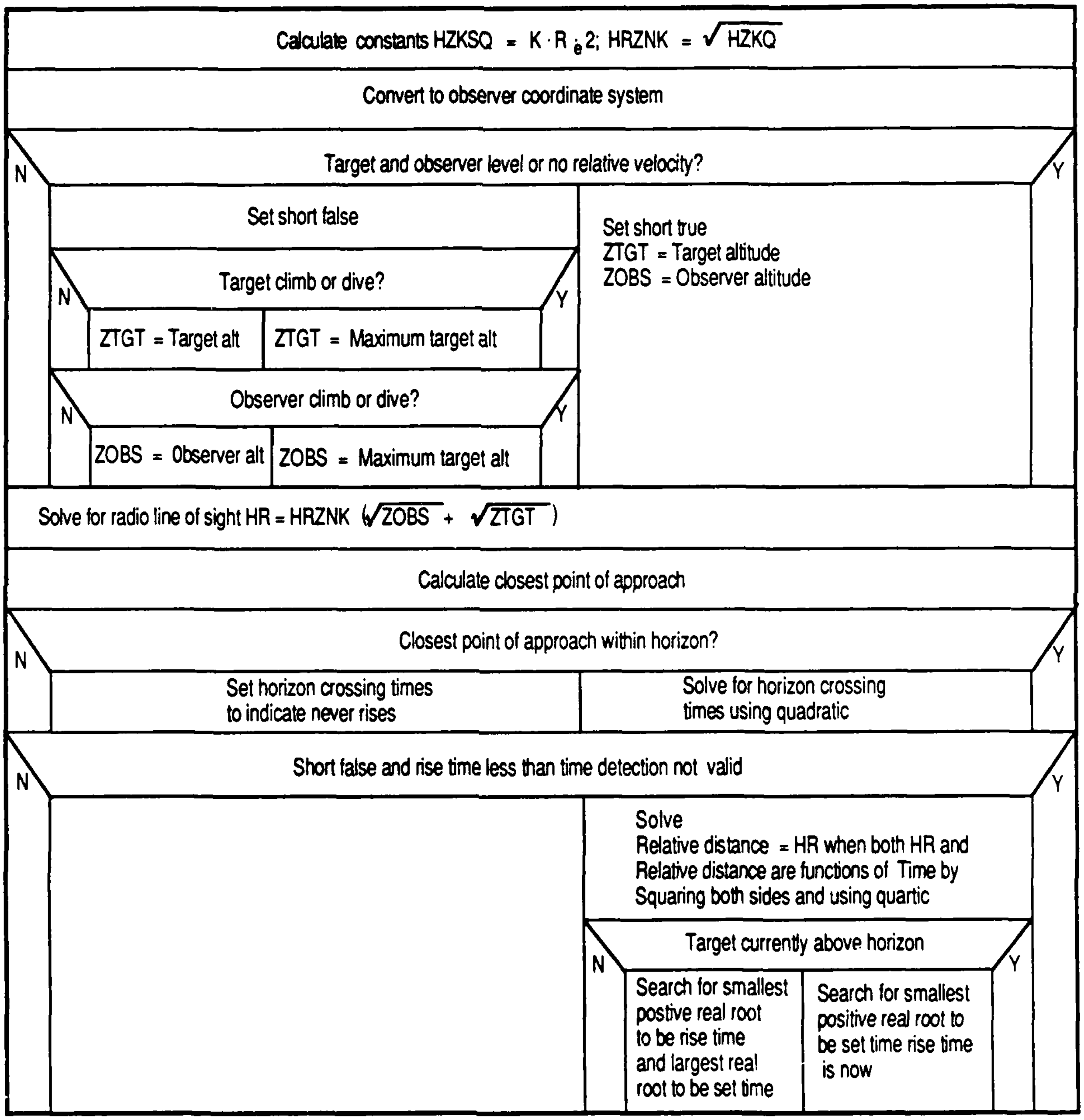


Figure 5-13 Flow for Subroutine XHOR

CHAPTER 9

SURFACE-TO-AIR-MISSILE SHIP MODULE

The Surface-to-Air-Missile (SAM) Ship Module simulates those ship activities for processing detected RED targets, and responding to force orders.

The targets are Air-to-Surface Missiles (ASMs), Surface-to-Surface Missiles (SSMs), Subsurface-to-Surface Missiles (SSMs), or enemy aircraft that venture closer to the defended point than the minimum range for interceptors.

9.1 BASIC SAM LOGIC

Four functions of the SAM ship are simulated:

- o decision;
- o fire control;
- o launch; and,
- o missile guidance.

In order to process a detected target, the ship must decide to engage. Then a missile must be removed from the magazine and mounted on a launcher. The launcher must be aimed, preflight tests and alignments conducted, and the missile launched. For the engagement to occur, the ships' fire control system must lock on the target and develope a solution. Finally the ship must provide guidance to the missile during the flight to the target.

9.1.1 Decision

The SAM ship module decides, for each target detected: whether the ship can engage the target; if the ship should engage the target; when to schedule the engagement based on earliest and latest time to intercept; and, with what type system, conventional or nuclear, to engage.

Ship Self-Assignment sectors are defined for each ship by the true bearing of the Counter Clockwise and the Clockwise sector limits defined in the BLUE ship characteristics common. Figure 9-1, SAM Ship Sectors, is an example of sector allocation. The command center may or may not be coordinating the battle. Note that the sectors can be defined to overlap one another.

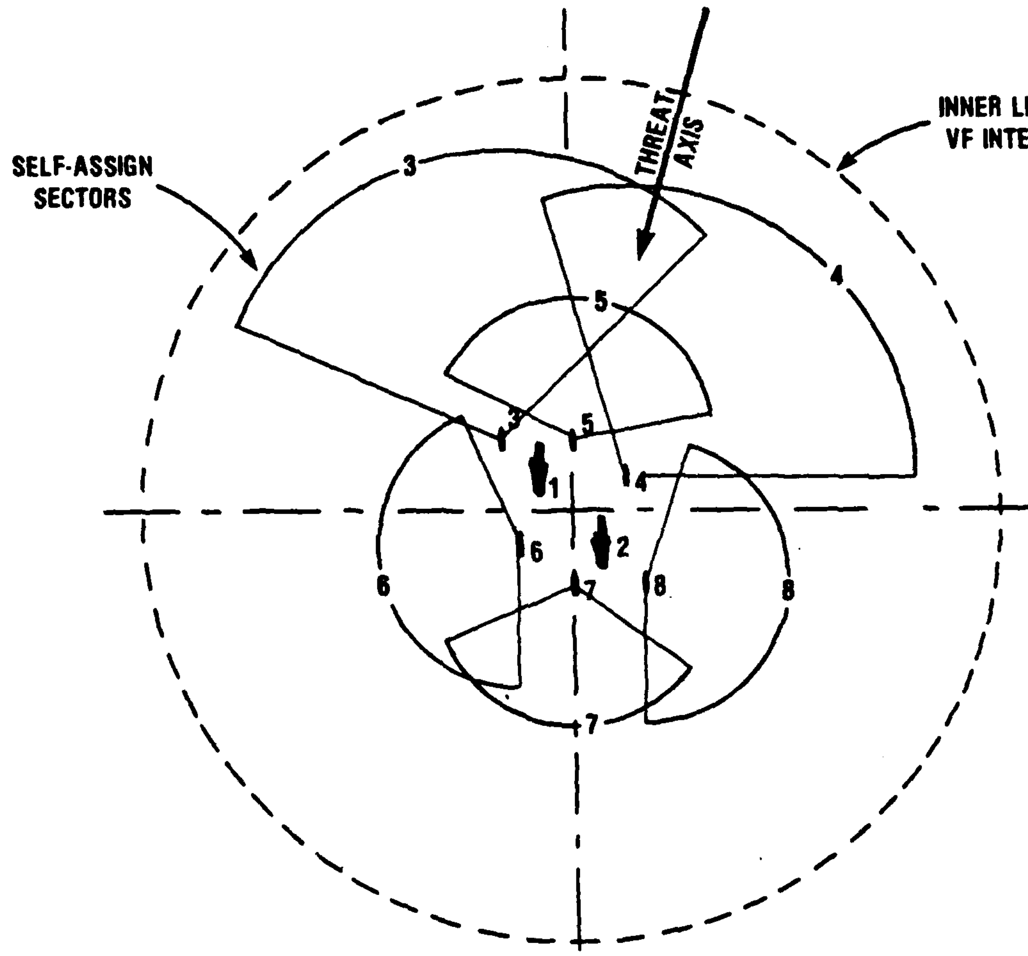


Figure 9-1, SAM Ship Sectors

9.1.1.1 Coordination

Three types of coordination are modeled and controlled by inputs to the model:

Type 1 : Command Center coordination

Type 2 : Sector coordination

Type 3 : No coordination

9.1.1.1.1 Command Center Coordination

The command center attempts to assign targets to individual ships in a more optimal fashion. To implement command center coordination, the concept of target priorities is introduced. Ten priorities of targets have been defined for the ship:

Priority 0 - Not a target for this ship.

Priority 1 - Target is a threat to own ship.
Employ nuclear SAM.

Priority 2 - Target is a threat to own ship.
Employ conventional SAM(s).

Priority 3 - Command center assigned target to ship.
Employ nuclear SAM.

Priority 4 - Command center assigned target to ship.
Employ conventional SAM(s).

Priority 5 - Ownership self-assignment to target.
unconfirmed.

Priority 6 - Threat to own ship and employing nuclear
SAM, above SAM ceiling

Priority 7 - Threat to own ship, above SAM ceiling

Priority 8 - Assigned by command center to employ nuclear
SAM, above SAM ceiling

Priority 9 - Command center assignment, above sam ceiling

Priority 10 - Self assignment, unconfirmed, above SAM ceiling

As each ship makes detections, it makes self-assignments to shoot at the targets. The ships will use sectors self-assignment sectors and perceived threats as decision criterion. These self-assignments are reported to the command center. The command center may make assignments to the individual ships or may negate sector self-assignments when notified. The highest priority target is one that is perceived as an immediate threat to own ship. Priority one targets not only go ahead of command center assignments, but they may also preempt or interrupt SAMs about to be launched against lower priority targets. Priority one targets may not be preempted. These concepts will be further developed in the following sections and in the description of the command center.

Assignments from the command center can be rejected by the SAM ship. At the time the command is received, subroutine TGTREJ (Target reject) is called to test if the ship:

1. is tracking the target or has the naval tactical data system (NTDS) available to receive information on the target,

2. has missiles available; or,

3. has at least one launcher channel and one fire control channel in up status.

If any of these tests fail, the assignment is immediately rejected. When the ship actually starts preparations to fire, the target is rejected at any point that the ship determines it can't get a needed resource in time to hit the target inside the SAM envelope.

9.1.1.1.2 Sector Coordination

The ships operate independently of the command center, but fire only at targets that are in their assigned sectors or are threats to own ship. This type of doctrine reduces the number of unengaged targets and overkills and thus makes better use of the SAMs available. The command center is not coordinating the battle.

9.1.1.1.3 No Coordination

Each ship may fire at any target it desires without regard to what other ships are doing. The command center is not directing the battle and there may be considerable overkill and unengaged targets.

9.1.1.2 SAM Doctrine

The number and type of SAM(s) fired at a target is based on the number and type of SAMs available, whether or not the force has nuclear release, the amount of time remaining before the target is too close to attempt another engagement of the target, and the SAM firing policy of the ship.

9.1.1.2.1 Firing Policies

Each SAM ship is assigned a SAM firing policy which is stored in the ship status common (SHSTAT). The SAM firing policy consists of two numbers. The first is the primary firing policy of the ship and the second number is the alternate firing policy which is executed when the primary firing policy can not be used due to a lack of a needed resource. There are eight SAM firing policies modeled which can accommodate either vertical launchers or trainable launchers. They are defined as follows:

LEGEND:

- S(n) - shoot a nuclear SAM.
- S - shoot a conventional SAM.
- SS - shoot shoot; that is, shoot two-missile salvo
- L - look, evaluation period.
- OS - O' boy, last firing opportunity, use last resort.

SAM firing policies:

1) S(n) - L - S(n) - L - - S(n)...
2) S - L - S - L - S - L - OS - SS
3) S - L - S - L - S - L - S...
4) SS - L - SS - L - SS - L - SS...
5) S - L - S - L - S - L - OS - S(n)
6) SS - L - SS - L - SS - L - OS - S(n)
7) S - L - S - L - S(n) - L - S(n)...
8) SS - L - SS - L - S(n) - L - S(n)...

NOTE: For SAM firing policies 7 and 8, the policy changes when the target comes within the nuclear keep out range from any ship in the force. The nuclear keep

out range is input for each type of target in the scenario.

9.1.2 Fire Control

The fire control model includes the time delay for search and acquisition by the fire control radar and the time to develop a target track. The GPSS seizes the next available fire control channel and assigns it to the target after the time the channel is required. After a time delay for the fire control channel to lock on and stabalize, the system is ready to process the target further. After the missile flight to the point of the intercept and a brief time for evaluation (to determine a hit or miss) the fire control channel can be released and made available for reuse. If a salvo launch was made, an inter-launch delay time is added to the time of availability for the fire control channel. A fire control channel can be preempted for a higher priority target prior to SAM launch. If not preempted, the fire control channel remains dedicated to a target until either target destruction or target intercept is no longer feasible.

9.1.3 Launch

A relatively large number of launcher configurations in the fleet impose the requirement for a flexible launcher model. There are three launcher types modeled: single rail, dual rail, and vertical. Single and dual rail launchers presently exist in the fleet while the vertical launching system is in development. The vertical launcher consists of fixed cells from which the SAMs are launched without being physically aimed. Therefore, the load time and the slew time will be zero for ships with vertical launch systems.

A brief survey of the fleet indicates ships with configurations as follows:

- One single rail;
- Two single rail;
- One dual rail;
- Two dual rail; and,
- One single rail and two dual rail.

Since the last configuration exists only on one ship, only the first four configurations are modeled, in addition to the vertical launching system. For the railed launch systems, a launcher will be selected provided one with a loaded SAM(s) is available. After a launcher slew delay time, the SAM is launched and the launcher is released to be reloaded.

9.1.4 Missile Guidance

The missile guidance model represents the means of guiding the SAM from the ship to the target. Three generic types of SAM guidance schemes have been identified for the simulation:

(1) Command all the way;

(2) Home all the way; and,

(3) Mid-course guidance and terminal homing.

At this time, schemes one and two are treated identically in the simulation; the guidance channels are combined with the fire control channels for schemes one and two. Figure 9-2, Functional flow with SAMTYP 1 or 2, shows the functional flow for a ship system using schemes one or two. The guidance/illuminator channel must be available from launch of the SAM until the time of the intercept when the channel can be released.

SAM guidance scheme three requires that the guidance channel, which represents the illuminator, be modeled as a separate element as shown in Figure 9-3, Functional Flow with SAMTYP 3 (Mid-course Guidance). The fire control channel must be available from launch of the SAM until terminal homing when an illuminator must be scheduled for the intercept.

The SAM ship event diagram in Figure 9-4, SAM Ship Events, is closely related to Figures 9-2, Functional flow with SAMTYP 1 or 2, and 9-3, Functional Flow with SAMTYP 3 (Mid-course Guidance) but is more GPSS oriented. The figure is appropriate for all three SAM guidance schemes. Also shown are the HELP blocks where the FORTRAN routines are used. It is through these HELP blocks that the workings of the simulation will be described.

9.2 DATA DEFINITIONS

The following are a list of the possible states that launchers can have:

Launcher load state. Stored in SSLLOD(BUID,LAUNCHER).

0 - Empty

1 - One conventional SAM loaded

2 - Two conventional SAMs loaded

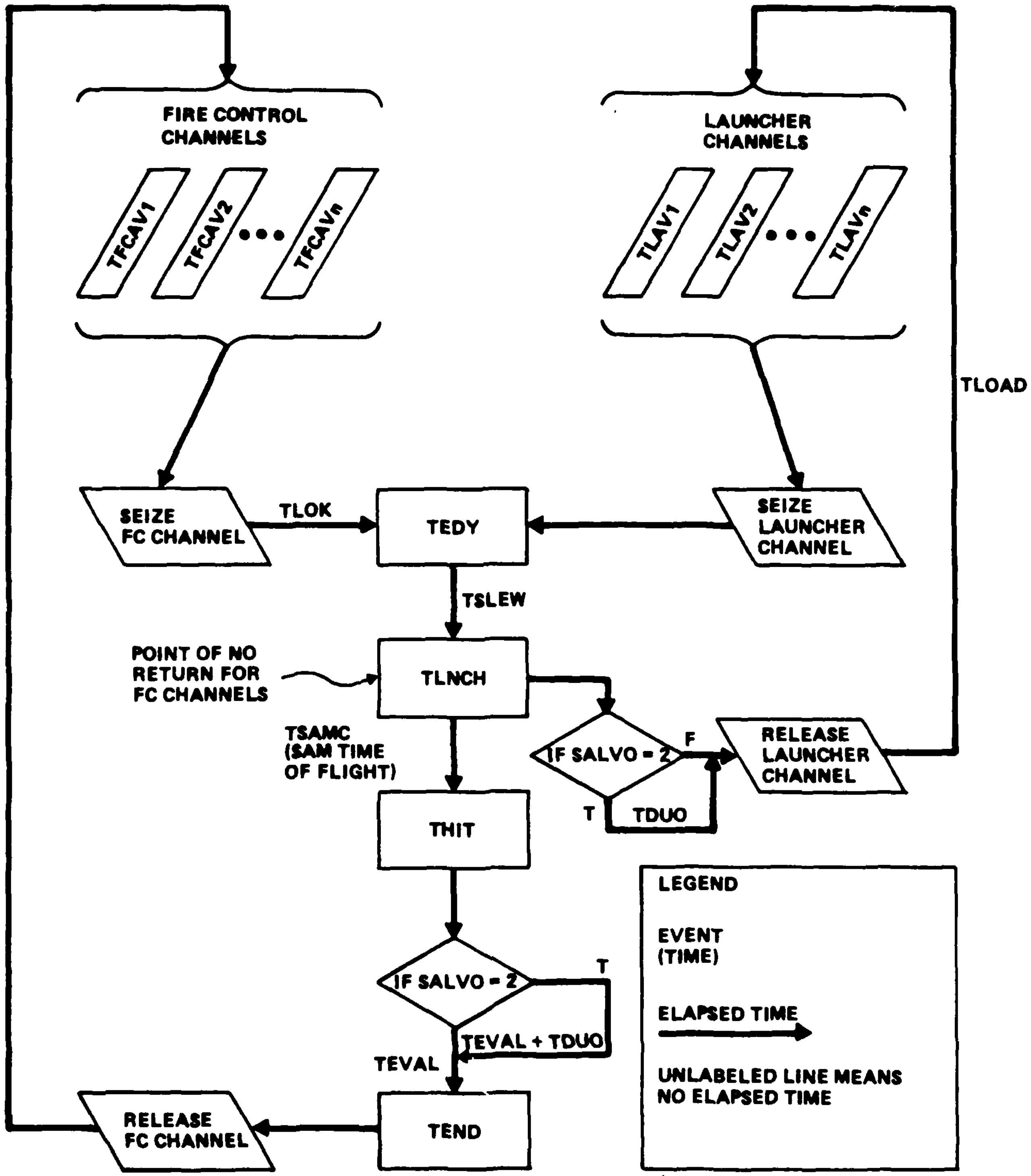


Figure 9-2. Functional Flow with SAMTYP 1 or 2
(Command all-the-Way or Home-all-the-Way)

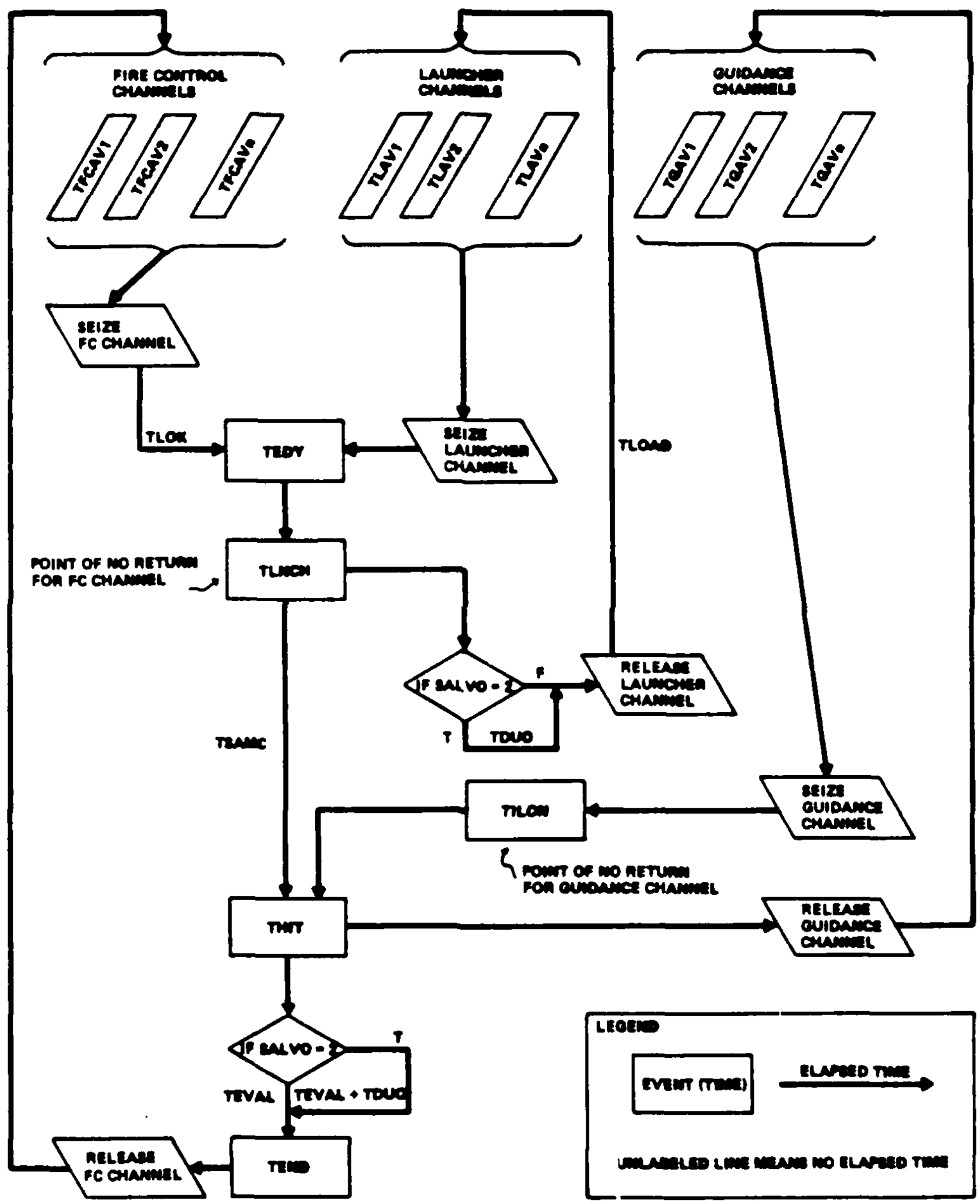


Figure 9-3. Functional Flow with SAMTYP 3 (Mid-Course Guidance)

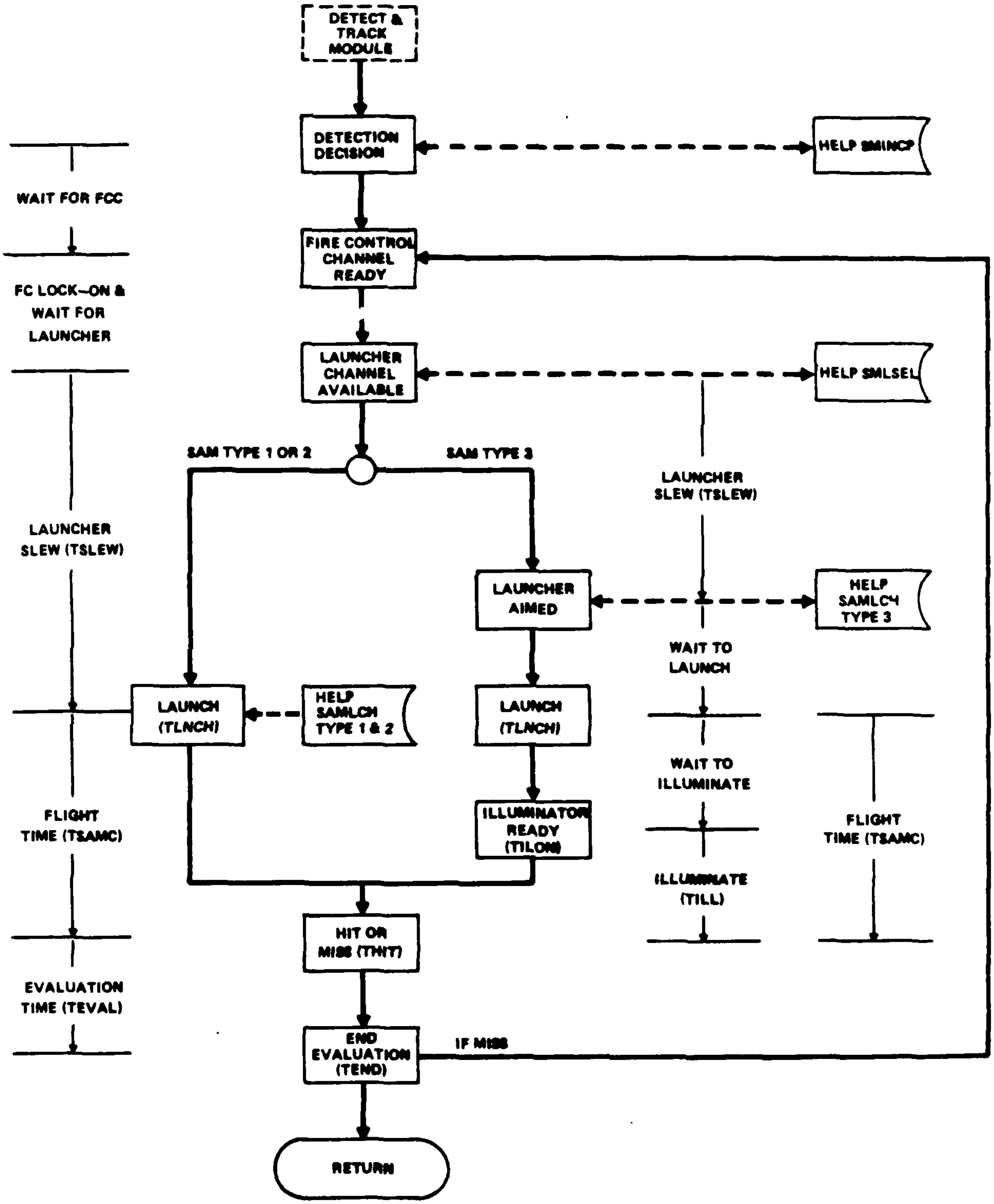


Figure 9-4. SAM Ship Events

21 - One nuclear SAM loaded

22 - Two nuclear SAMs loaded

23 - One conventional, one nuclear SAM loaded

Launcher state. Stored in SSLSTA(BUID,LAUNCHER). (Being loaded means missiles are being removed from the magazine and placed on the launcher. Ready means the launcher is loaded with a missile. In slew means that the missiles have been loaded and the launcher is being aimed.)

0 - Inoperable

11 - Ready one SAM

12 - Ready two SAMs

21 - In slew, preemptable, one SAM

22 - In slew, preemptable, two SAMs, two assigned

23 - In slew, preemptable, 2 SAMs, one assigned

31 - In slew, not preemptable, 1 SAM

32 - In slew, not preemptable, 2 SAMs,
two assigned

33 - In slew, not preemptable, 2 SAMs,
one assigned

41 - Being loaded, conventional, 1 rail

42 - Being loaded, conventional, 2 rails

43 - Being loaded, nuclear, 1 rail

44 - Being loaded, nuclear, 2 rails

The following is a list of the possible values for the engagement status for a target. Twenty different status conditions have been defined for the ship-target pair:

0 - Not processed

1 - Target engageable

2 - In queue for fire control channel

3 - Have fire control channel - locking on

4 - In queue for launcher channel

5 - One launcher channel - in slew

6 - Two launcher channels - in slew

7 - In illuminator queue - one launcher channel

8 - In illuminator queue - two launcher channels

9 - In illuminator queue - vertical launcher

10 - Ready to fire - one launcher channel

11 - Ready to fire - two launcher channels

12 - Ready to fire - vertical launcher

13 - Firing, past point of no return, one Launcher Channel

14 - Firing, past point of no return, two Launcher Channels

15 - Firing, vertical launcher

20 - Not engageable, or cancelled by Command Center

21 - Not in sector

22 - Engaged by another ship

23 - Waiting for answer to nuclear self-assign request

9.3 FORTRAN SUBPROGRAMS REQUIRED

The following is a list of the FORTRAN subprograms required to operate the SAM ship module. Figure 9-5, FORTRAN Subroutine Usage, shows the routines used and the calling routines.

INCPT — Computes the point and time that a SAM will (if ever) intercept a target. Inputs to this routine are the speeds and coordinates of the target and the speeds and coordinates of the intercepting SAM. Outputs are the time to the intercept (which must be added to the current clock time to get the time of the next event), the distance from the launching ship to the intercept point, the coordinates of the intercept, and a flag which indicates whether an intercept is possible.

LNCHSL — Selects an available or preemptable launcher by comparing the requirements for launchers (generated by the tactical situation and the firing policy) with the launcher status and loading.

MKMSG — Structures a message which will be sent to the GPSS module.

MOVEL — Updates the list of the moving BLUE aircraft and/or RED units. The amount of fuel remaining in BLUE fighters is also updated to the current clock time.

SAMCAN — Cancels the assignment of SAM ship resources to a given RED target.

SAMCMP — Interface routine between the RED threat module and SAMENV. The SAM envelope entry and exit times are calculated for each target leg and stored in the SSSTC1 and SSSTC2 data arrays in SHSTAT common for later use in the command module and in the SAM ship module. Note: this is done in the RED threat module to avoid multiple calculations.

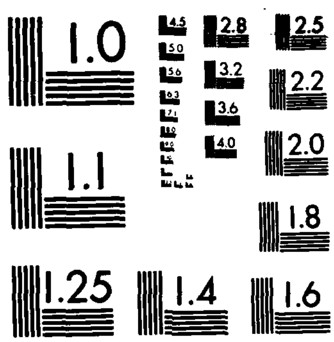
SAMENV — Computes the times and positions at which a RED target enters and leaves the SAM envelope of a given ship.

SAMFP1 — Defines the first SAM firing policy.
S(n) - L - S(n) - L - S(n) - L - S(n)...

SAMFP2 — Defines the second SAM firing policy.
S - L - S - L - S - L - OS - SS

AD-A174 818 DESIGN NOTEBOOK FOR NAVAL AIR DEFENSE SIMULATION (NADS) 3/3
REVISION(U) TRW DEFENSE SYSTEMS GROUP MCLEAN VA
WATERWHEEL PROGRAM OFFICE 09 OCT 86 N00014-83-C-0527

UNCLASSIFIED F/G 15/3 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Called \ Caller	HLPRTN	SAMENV	SAMFP1	SAMFP2	SAMFP3	SAMFP4	SAMFP5	SAMFP6	SAMFP7	SAMFP8	SAMGT3	SAMLCH	SMINCP	SMLORD	SMLSEL
INCPT			●	●	●	●	●	●	●	●		●		●	
LNCHSL															●
MKMSG				●	●	●	●	●				●	●		
MOVEL	●											●	●		●
SAMCAN	●														
SAMENV													●		
SAMFP1												●			●
SAMFP2												●			●
SAMFP3												●			●
SAMFP4												●			●
SAMFP5												●			●
SAMFP6												●			●
SAMFP7												●			●
SAMFP8												●			●
SAMGT3												●			
SAMLCH	●														
SMINCP	●														
SMLOAD	●														
SMLORD												●			
SMLSEL	●														
TGTREJ	●														
FUNCTION XYDIST		●							●	●	●		●	●	

Figure 9-5 FORTRAN Subroutine Usage

SAMFP3 Defines the third SAM firing policy.
S - L - S - L - S - L - S - L -...

SAMFP4 Defines the fourth SAM firing policy.
SS - L - SS - L - SS - L - SS - L ...

SAMFP5 Defines the fifth SAM firing policy.
S - L - S - L - S - L - OS - S(n)

SAMFP6 Defines the sixth SAM firing policy.
SS - L - SS - L - SS - L - OS - S(n)

SAMFP7 Defines the seventh SAM firing policy. The policy is changed when the target comes within the nuclear keep out range of any ship in the force.
S - L - S - L - S - L - S(n) - L - S(n) - L ...

SAMFP8 Defines the eighth SAM firing policy. The policy is changed when the target comes within the nuclear keep out range of any ship in the force.
SS - L - SS - L - SS - L - S(n) - L - S(n) - L ...

SAMGT3 Schedules a SAM illuminator for semi-active terminal guidance. The time of the launch is returned.

SAMLCH Called when a SAM is scheduled to be launched.

SMINCP Determines the next action to be taken by the BLUE ship in response to a RED target detection or course change within the SAM envelope. The priority of the target is also determined.

SMLOAD Reloads one or two empty single or dual rail launchers and marks them ready for use.

SMLORD Determines the load order to be given. The order is stored in the SAM ship launcher state array.

SMLSEL Calculates the number of SAMs available, determines the possiblity of intercept with the type of SAMs available, and selects an appropriate launcher (via the launcher selection routine LNCHSL).

TGTREJ Permits a SAM ship to reject a target when the ship is not tracking the target or the ship does not have an operational Naval tactical data system or the ship does not have the needed resources to engage the target.

Function
XYDIST Computes the two dimensional distances between two points.

9.3.1 Subroutine LNCHSL

Subroutine LNCHSL contains the launcher selection logic. LNCHSL is called by SMLSEL. Selection of an appropriate launcher involves comparing the needs of the ship's firing policy (will two rails be needed?, will a nuclear warhead be needed?) and the priority of the target with what the launcher(s) presently has loaded and to what targets, if any, are the loaded SAMs assigned. Output from this routine is the launcher selected, the RED ID of the the preempted target(s), and the SAM status which will be either one or two launcher channels in slew (SMSTAT is 5 or 6). The first launcher that has the required and available SAMs will be selected. If the target is a priority one or five and all available launchers are already assigned to targets, then the first launcher that has the needed SAMs which is not assigned to a a target that is a threat to own ship is preempted. The complete logical flow is shown in Figure 9-6, Flow Diagram for Subroutine LNCHSL.

9.3.2 Subroutine SAMENV

SAMENV contains the SAM envelope intersection logic. SAMENV is called by SMINCP. Subroutine SAMENV computes the time and location at which enemy aircraft or missiles enter and leave the SAM envelope of a particular ship.

To compute these intersection points, the position and speed of the RED target are transferred from the X-Y coordinate system to a U-V coordinate system. The V direction is parallel to but opposite in direction to the path of the incoming RED target as shown in Figure 9-7, Alignment of SAM Envelope to Target.

LNCHSL

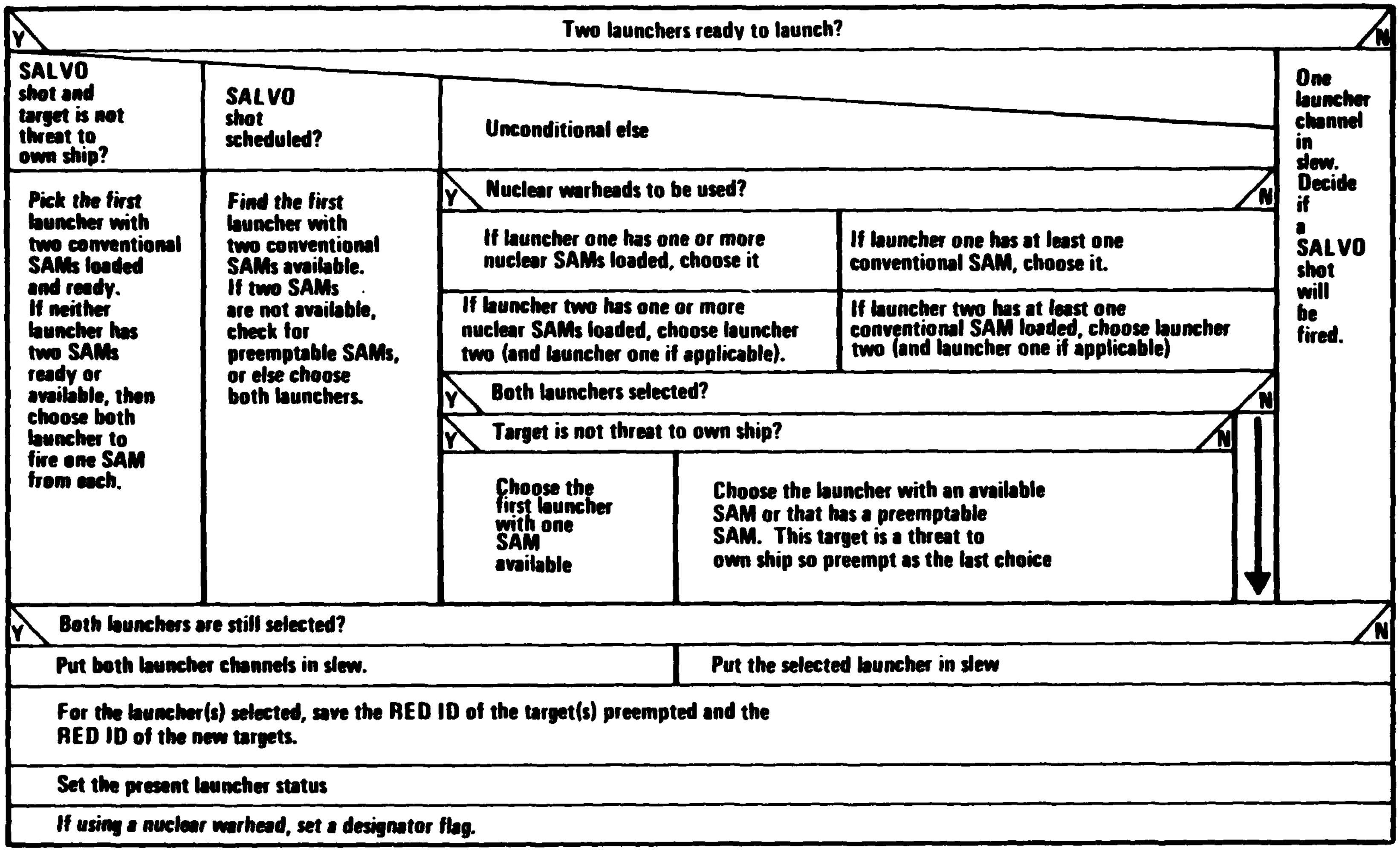


Figure 9-6 Diagram for Subroutine LNCHSL

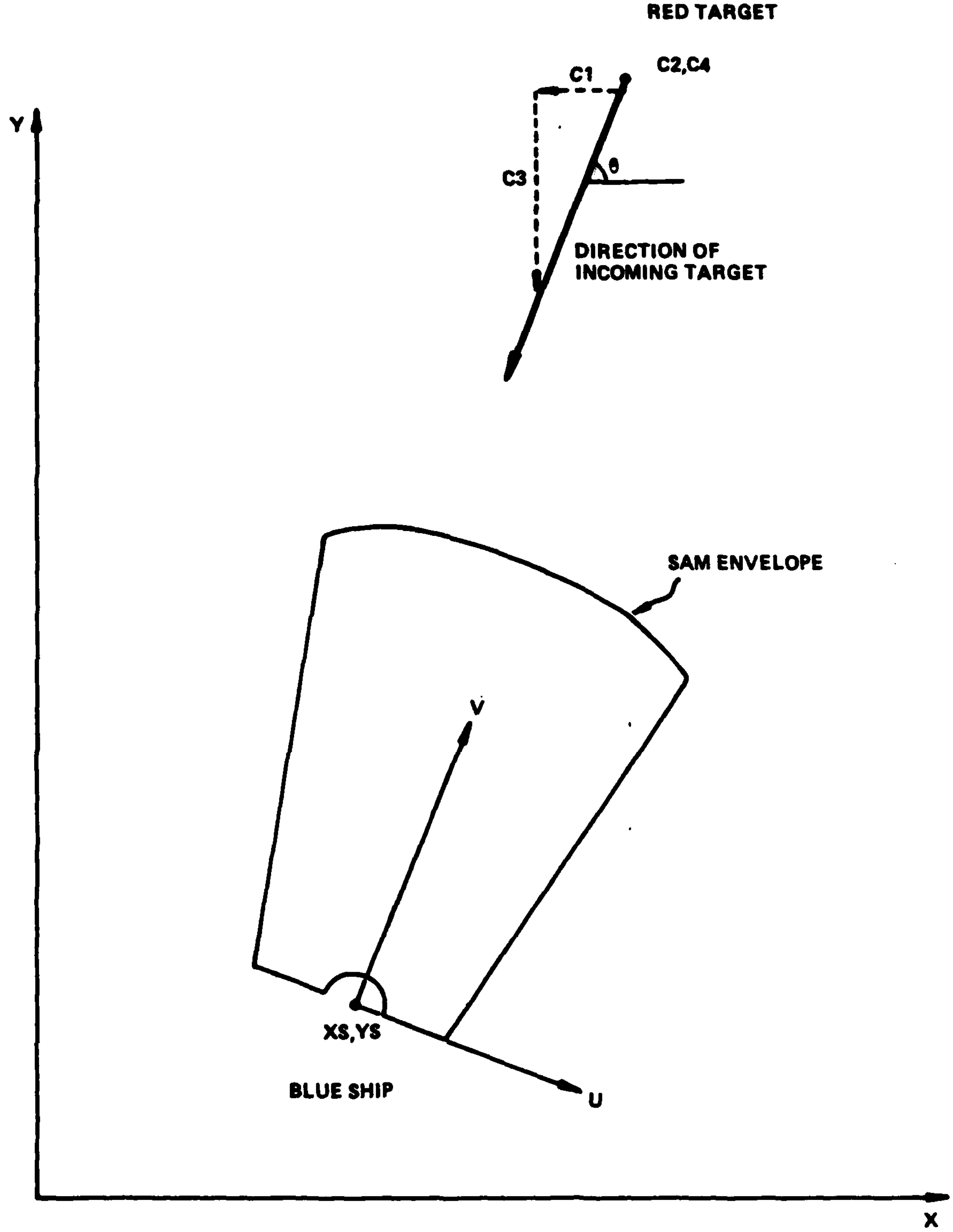


Figure 9-7 Alignment of SAM Envelope to Target

The variable names for the RED and BLUE units are given below:

C1 - X-speed component of RED unit

C2 - X-coordinate of RED unit

C3 - Y-speed component of RED unit

C4 - Y-coordinate of RED unit

XS - X-coordinate of BLUE ship

YS - Y-coordinate of BLUE ship

theta - Angle that the path of the RED unit makes with the X-axis

U1 - U-coordinate of RED unit

V1 - V-coordinate of RED unit

The transformation equations from the X-Y to the U-V coordinate system are shown below where S1 through S8 are coefficients based upon the direction of the incoming target.

U1 = S1 * (C2-XS) * SIN(theta)+ S2 * (C4-YS) * COS(theta)

V1 = S3 * (C2-XS) * COS(theta)+ S4 * (C4-YS) * SIN(theta)

The inverse transformation is given by:

C2 = S5 * U1 * SIN(theta) + S6 * V1 * COS(theta) + XS

C4 = S7 * U1 * COS(theta) + S8 * V1 * SIN(theta) + XS

By requiring that the V direction be parallel to the direction of the incoming target, the U coordinate of the intercept will be the same whether the target is entering or leaving the envelope. Once the U coordinate (U1=U2) is calculated, it can be used to calculate the V coordinate of the intercept for both intersection points.

The basic SAM envelope configuration is shown in Figure 9-8, Typical SAM Envelope. The front portion (Zone 1) consists of a segment of a super ellipse given by the equation:

(U1 / A)** N + (V1 / B)** N = C

Where A, B, C and N are coefficients that fit the super ellipse to the SAM

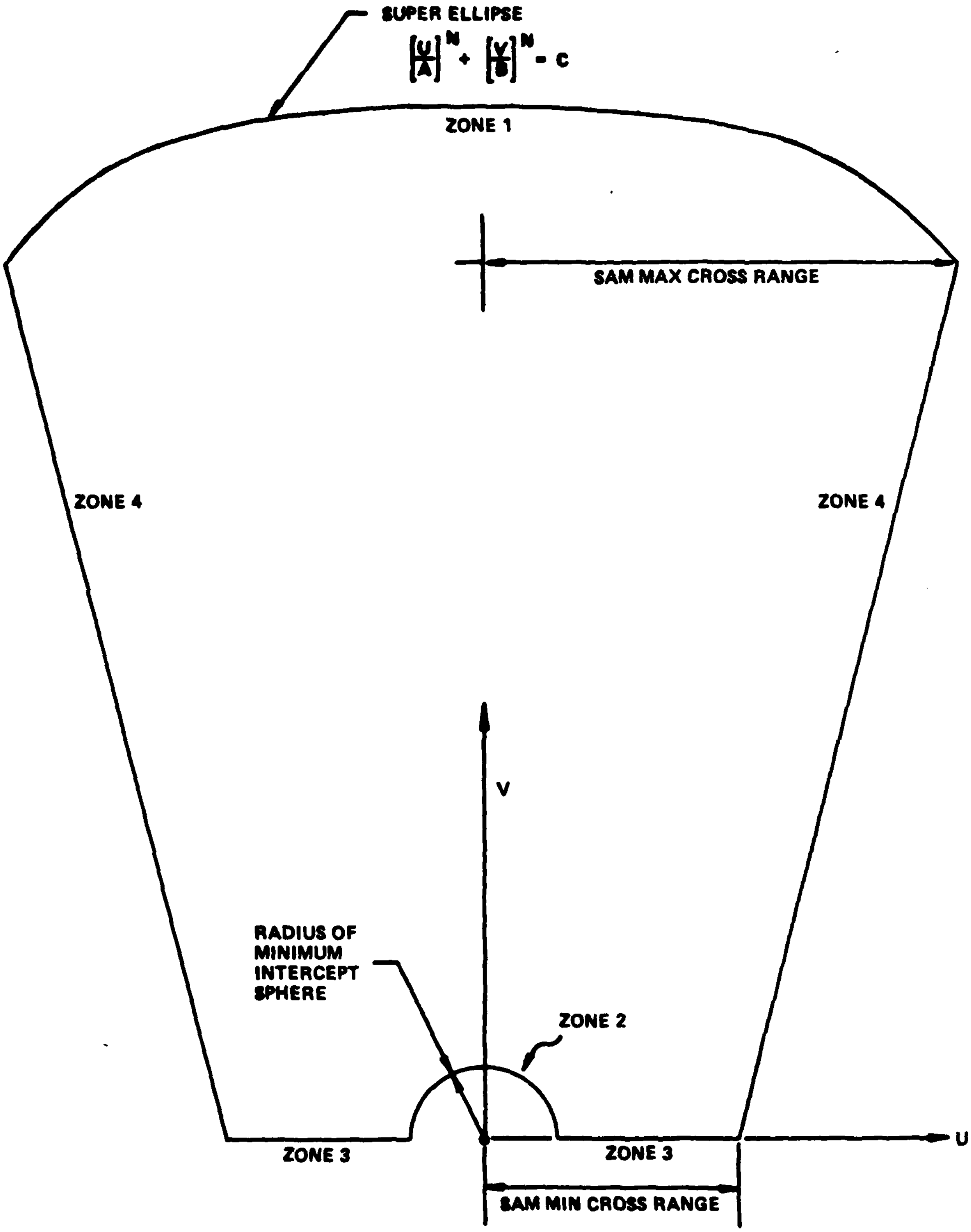


Figure 9-8 Typical SAM Envelope

envelope. The rear portion of the envelope consists of a minimum intercept sphere (Zone 2) closest to the ship which connects to a line at V=0. This line (Zone 3) stops at the minimum cross range. The envelope is completed by connecting the minimum cross range at V=0 to the point where the maximum cross range cuts off the super ellipse (Zone 4).

Each SAM envelope can be defined by six variables:

SM1 - SAM maximum cross range

SM2 - SAM minimum cross range

SM3 - Coefficient of the super ellipse equation

SM4 - Coefficient of the super ellipse equation

SM5 - Coefficient of the super ellipse equation

SM6 - Coefficient of the super ellipse equation

These values will be stored in the SAM missile characteristics common (SMCHAR).

Subroutine SAMENV first calculates U1, the U-coordinate of the intercept. Once U1 is known it can be substituted into the super ellipse equation to find the V coordinate of the first intercept. The logic then determines which of the three remaining zones the target will intercept and calculates the V coordinate of the second intercept. After the coordinate transformation, the distances between each of these two points and the current position of the RED unit are then calculated. Using these distances and the velocity of the target, the associated times to reach these points are subsequently computed and returned. The complete logical flow is shown in Figure 9-9, Flow Diagram for Subroutine SAMENV.

9.3.3 SAM Firing Policies

Presently there are eight SAM firing policies to be used by the SAM ships. These firing policies can be used for vertical as well as trainable launchers. The only real difference in these routines for either type of launcher is in the way the time of the intercept is calculated. For trainable launchers, slew time will be added to the time of the intercept, but for vertical launchers, the slew time will always be zero. These routines are called from SAMLCH (SAM launch) for vertical launchers and from SMLSEL (SAM missile) for trainable launchers.

SAMENV

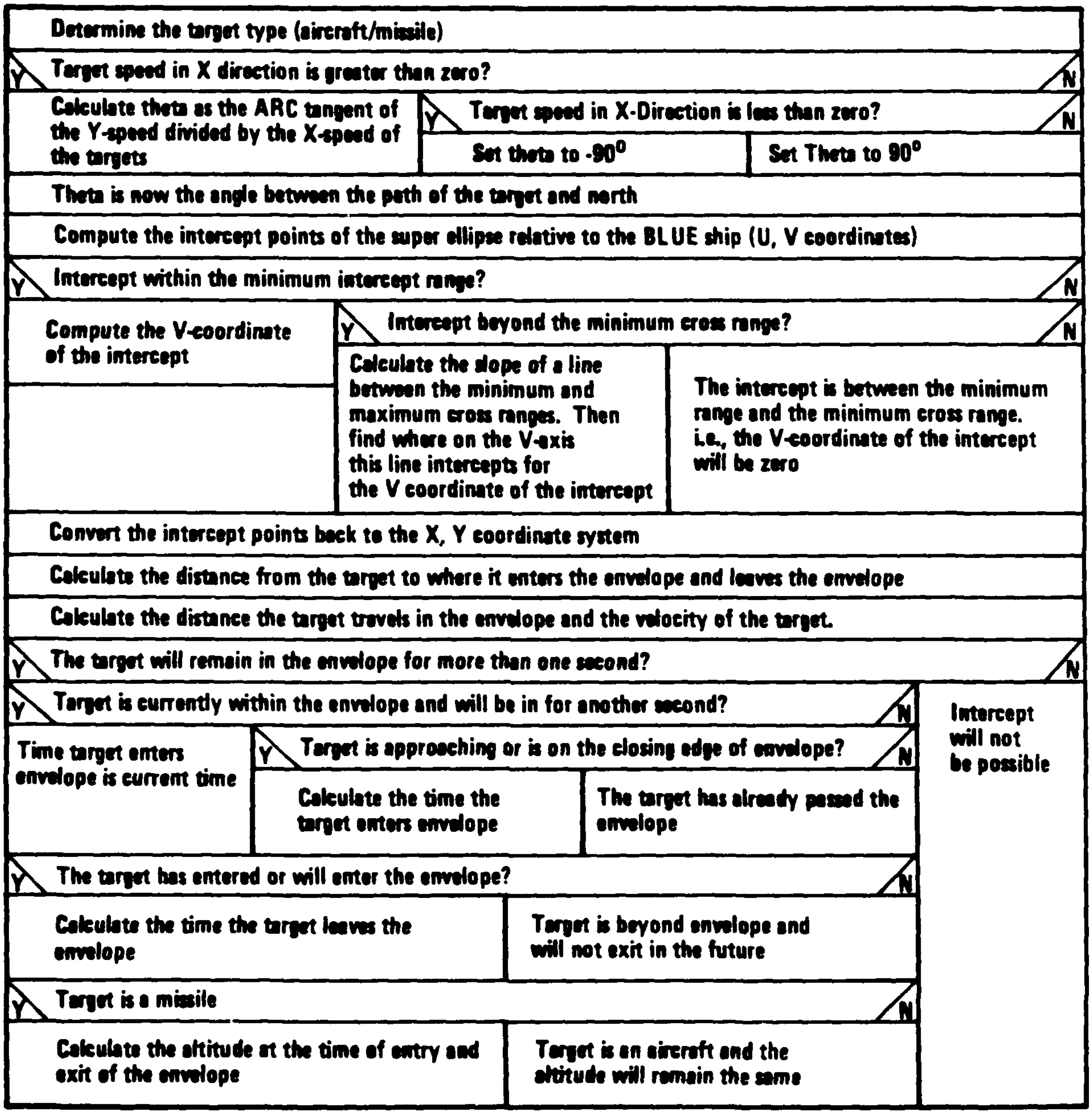


Figure 9-9 Flow Diagram for Subroutine SAMENV

9.3.3.1 Subroutine SAMFP1

Subroutine SAMFP1 contains the logic to support SAM firing policy one;

S(n) - L - S(n) - L - S(n) - L - S(n) - L ....

Firing policy one is a preferred nuclear firing policy. This routine will be called only if the force has nuclear release (logical variable NUCREL stored in the miscellaneous common) and there are nuclear SAMs available.

SAM firing policy one will check the possibility of a nuclear intercept via the INCPT routine. If a nuclear intercept is possible, the time of the nuclear intercept is calculated and the warhead type is set to indicate a nuclear missile.

If a nuclear intercept is not possible, the possibility of a conventional intercept will be checked. If a conventional intercept is possible, then the warhead type will be conventional, the time of conventional intercept will be calculated, and a two-missile salvo shot will be made if two SAMs are available and the evaluation period will extend past the last opportunity for an intercept attempt.

When a nuclear and conventional intercept is not possible, the warhead type will be set to zero. The SAM state will indicate that the target is not engageable, and the next event will be a loss of detection which means that no further processing of this target will be made by this SAM ship.

When used as a primary firing policy, the alternate firing policy must be two, three, or four (SAMFP2, SAMFP3, or SAMFP4). The complete logical flow is shown in Figure 9-10, Flow Diagram for Subroutine SAMFP1.

9.3.3.2 Subroutine SAMFP2

Subroutine SAMFP2 contains the logic to support SAM firing policy two;

S - L - S - L - S - L - OS - SS

SAM firing policy two prefers single conventionally armed missiles. This routine will be called only when the input firing policy of the ship is two and there are conventional SAMs available.

The possibility of a conventional intercept is calculated via the INCPT routine. If a conventional intercept is possible, the time of the intercept is calculated and the warhead type is set to indicate a conventional SAM. A two-missile salvo will be made if there is not enough time to evaluate the results of this shot and still fire another conventional SAM at the target (this is the last firing opportunity).

SAMFP1

- **The force has nuclear release and nuclear SAMS are available?**
  - Y:
    - calculate the time, place and possibility of a nuclear intercept via INCPT routine.
    - **Nuclear intercept possible?**
      - Y:
        - Calculate time of the intercept. Set warhead type to nuclear
      - N:
        - Calculate the time, place and possibility of a conventional intercept via INCPT routine
        - **Conventional intercept possible?**
          - Y:
            - Calculate the time of the conventional intercept.
            - Set warhead type to conventional
            - Shoot SALVO if there is not enough time for another launch after this and there are conventional SAMs left.
          - N:
            - Neither a conventional or nuclear intercept is possible.
            - Next event is loss of detection
  - N

Figure 9-10 Flow Diagram for Subroutine SAMFPI

When a conventional intercept is not possible, the possibility of a nuclear intercept is calculated provided there are nuclear SAMs available and the force has nuclear release. If the nuclear intercept is possible and the target is priority one, the warhead type will be marked nuclear and the time of the intercept calculated. When the target is not a priority one, a message is sent to the command center asking for permission to engage the target with a nuclear SAM. The SAM ship will wait for a response from the command center whether to engage the target with nuclear SAMs.

When a conventional and nuclear intercept is not possible, the warhead type will be set to zero, the SAM state will indicate that the target is not engageable, and the next event will be a loss of detection which means that no further processing of this target will be made by this ship. The complete logical flow is shown in Figure 9-11, Flow Diagram for Subroutine SAMFP2.

9.3.3.3 <u>SubroutineSAMFP3</u>

Subroutine SAMFP3 contains the logic to support SAM firing policy three;

S - L - S - L - S - L - S - L ....

SAM firing policy three is the second of three SAM firing policies which prefer conventionally armed missiles. This routine will be called only when the input firing policy of the ship is three and there are conventional SAMs available.

The possibility of a conventional intercept is calculated via the INCPT routine. If a conventional intercept is possible, the time of the intercept is calculated and the warhead type is set to indicate a conventional SAM. A two-missile salvo will be made if there is not enough time to evaluate the results of this shot and still fire another conventional SAM at the target (this is the last firing opportunity).

When a conventional intercept is not possible, the possibility of a nuclear intercept is calculated provided there are nuclear SAMs available and the force has nuclear release. If the nuclear intercept is possible and the target is priority one, the warhead type will be marked nuclear and the time of the intercept calculated. When the target is not a priority one, a message is sent to the command center asking for permission to engage the target with a nuclear SAM. The SAM ship will wait for a response from the command center whether to engage the target with nuclear SAMs.

When a conventional and nuclear intercept is not possible, the warhead type will be set to zero, the SAM state will indicate that the target is not engageable, and the next event will be a loss of detection which means that no further processing of this target will be made by this ship.

SAMFP2

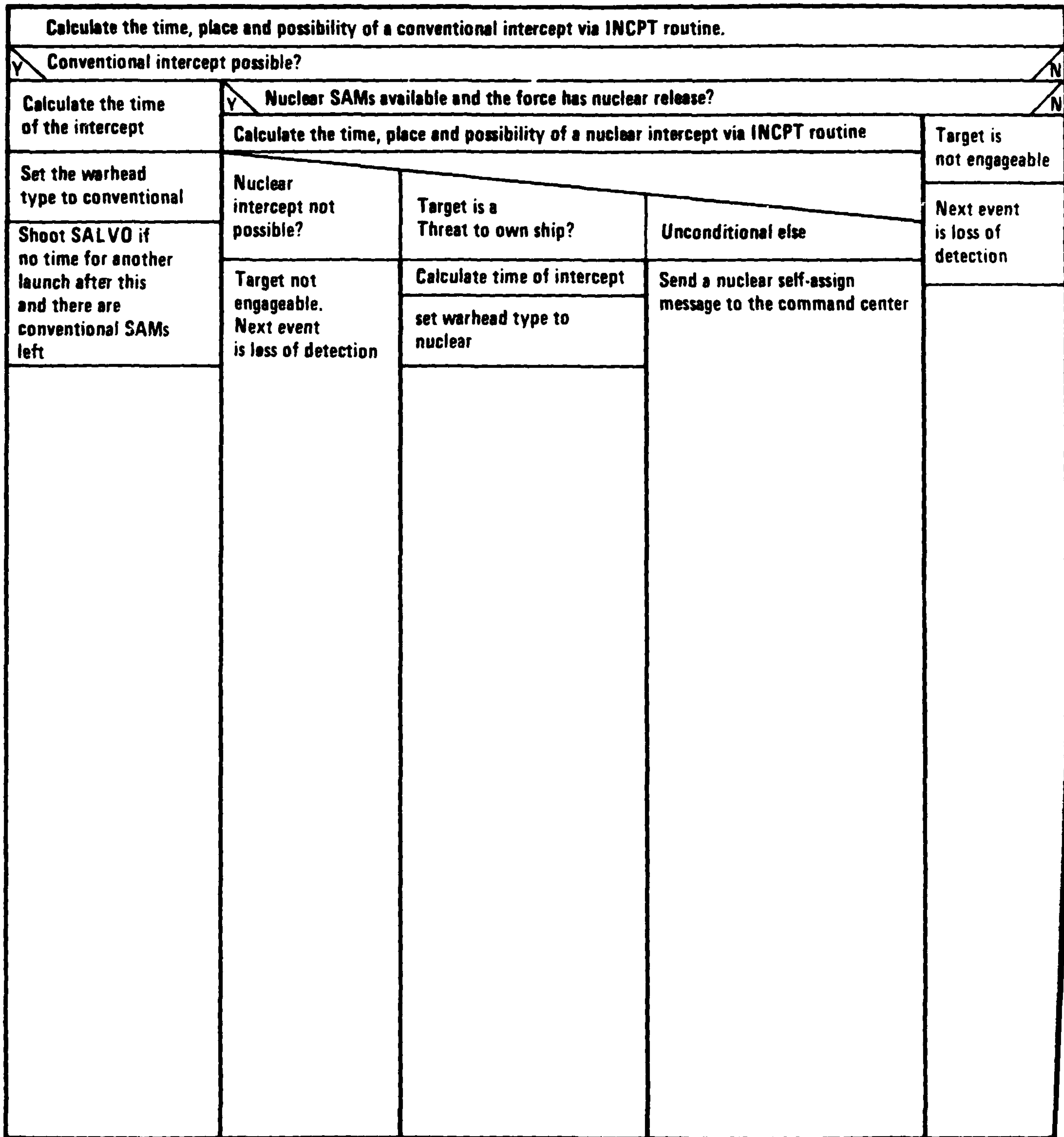


Figure 9-11 Flow Diagram for Subroutine SAMFP2

SAM firing policy three differs from policy two only in that three lacks a change in salvo size for the last firing opportunity. The complete logical flow is shown in Figure 9-12, Flow Diagram for Subroutine SAMFP3.

9.3.3.4 Subroutine SAMFP4

Subroutine SAMFP4 contains the logic to support SAM firing policy four;

SS - L - SS - L - SS - L - SS - L - ...

SAM firing policy four is the last SAM firing policy which prefers conventionally armed missiles . This routine will be called when the firing policy of the ship is four and there are conventional SAMs available.

When a target has a priority of four or five, the primary and alternate firing policies are set to one and four respectively overriding any input firing policy for this launch only.

The possibility of a conventional intercept is calculated via the INCPT routine. If a conventional intercept is possible, the time of the intercept is calculated and the warhead type is set to indicate a conventional SAM. A two-missile salvo will be made if there is not enough time to evaluate the results of this shot and still fire another conventional SAM at the target (this is the last firing opportunity).

When a conventional intercept is not possible, the possibility of a nuclear intercept is calculated provided there are nuclear SAMs available and the force has nuclear release. If the nuclear intercept is possible and the target is priority one, the warhead type will be marked nuclear and the time of the intercept calculated. When the target is not a priority one, a message is sent to the command center asking for permission to engage the target with a nuclear SAM. The SAM ship will wait for a response from the command center whether to engage the target with nuclear SAMs.

When a conventional and nuclear intercept is not possible, the warhead type will be set to zero, the SAM state will indicate that the target is not engageable, and the next event will be a loss of detection which means that no further processing of this target will be made by this ship.

Firing policies four and two differ only in that four will fire a conventional two-missile salvo at every opportunity. The complete logical flow is shown in Figure 9-13, Flow Diagram for Subroutine SAMFP4.

SAMFP3

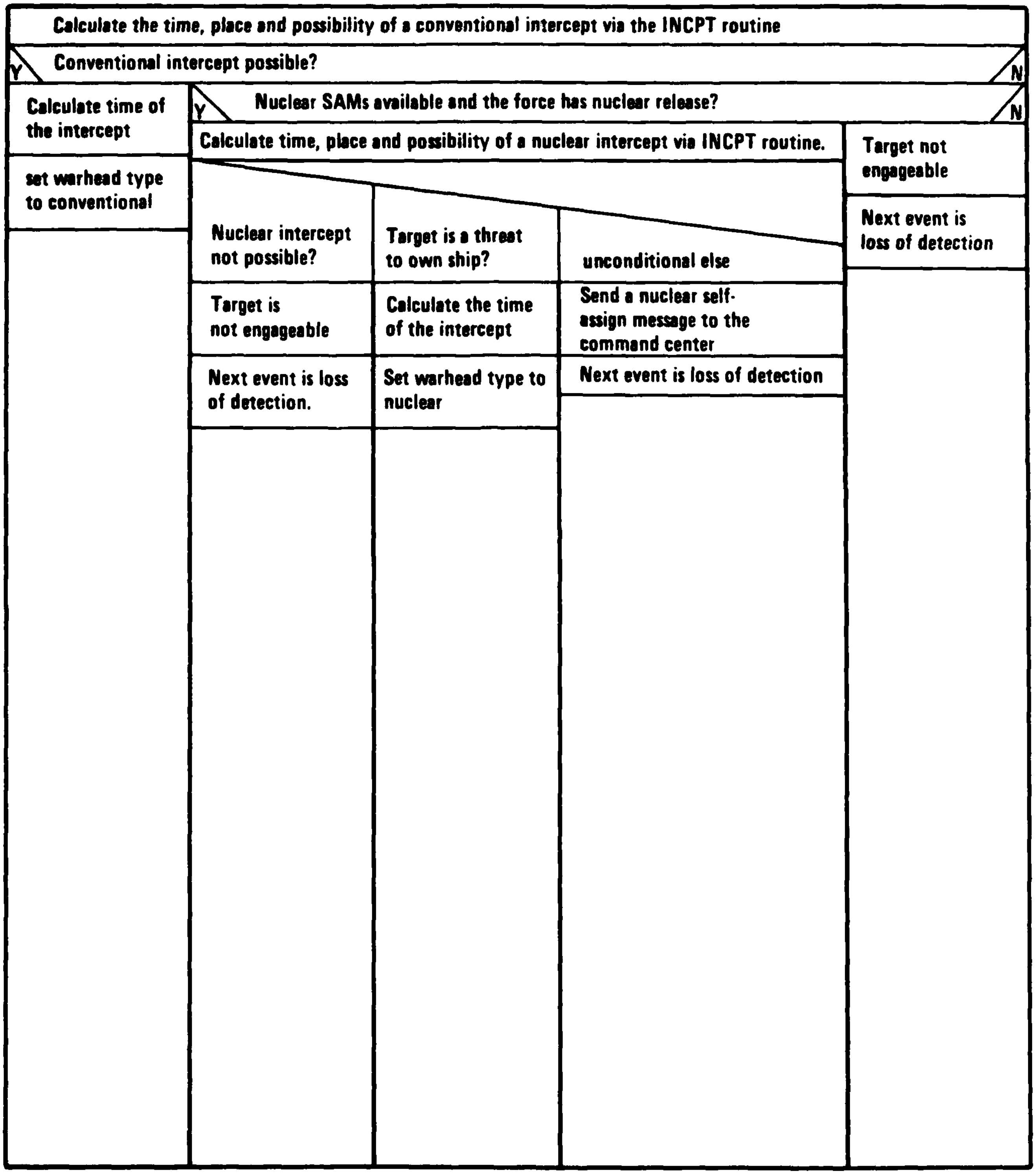


Figure 9-12 Flow Diagram for Subroutine SAMFP3

SAMFP4

Calculate the time, place and possibility of a conventional intercept via INCPT routine.

Conventional intercept possible?

- **Y**
  - Calculate the time of intercept.
  - Set warhead type to conventional.
  - Shoot SALVO if two or more conventional SAMs are remaining.
- **N**
  - Nuclear SAMs available and the force has nuclear release?
    - **Y**
      - Calculate time, place, and possibility of a nuclear intercept via INCPT routine
      - Nuclear intercept not possible?
        - Target not engageable.
        - Next event is loss of detection
      - Target is a threat to own ship?
        - Calculate time of intercept.
        - Set the warhead type to nuclear
      - Unconditional else
        - Send a nuclear self assign message to the command center.
        - Next event is loss of detection
    - **N**
      - Target is not engageable
      - Next event is loss of detection

Figure 9-13 Flow Diagram for Subroutine SAMFP4

9.3.3.5 Subroutine SAMFP5

Subroutine SAMFP5 contains the logic to support SAM firing policy five;

S - L - S - L - S - L - S - L - OS - S(n)

SAM firing policy five prefers single salvos of conventionally armed missiles until the last firing opportunity when a nuclear weapon will be selected. SAM firing policy five will be called only when the input firing policy of the ship is five and there are conventional SAMs available.

The possibility of a conventional intercept is calculated via the INCPT routine. If a conventional intercept is possible, the time of the intercept is calculated and the warhead type is set to indicate a conventional SAM. A two-missile salvo will be made if there is not enough time to evaluate the results of this shot and still fire another conventional SAM at the target (this is the last firing opportunity).

When a conventional intercept is not possible, the possibility of a nuclear intercept is calculated provided there are nuclear SAMs available and the force has nuclear release. If the nuclear intercept is possible and the target is priority one, the warhead type will be marked nuclear and the time of the intercept calculated. When the target is not a priority one, a message is sent to the command center asking for permission to engage the target with a nuclear SAM. The SAM ship will wait for a response from the command center whether to engage the target with nuclear SAMs.

When a conventional and nuclear intercept is not possible, the warhead type will be set to zero, the SAM state will indicate that the target is not engageable, and the next event will be a loss of detection which means that no further processing of this target will be made by this ship.

Firing policies five and two differ only in that five will fire a nuclear SAM at the last firing opportunity. The complete logical flow is shown in Figure 9-14, Flow Diagram for Subroutine SAMFP5.

9.3.3.6 Subroutine SAMFP6

Subroutine SAMFP6 contains the logic to support SAM firing policy six;

SS - L - SS - L - SS - L - SS - L - OS - S(n)

SAM firing policy six prefers salvos of two conventionally armed missiles until the last firing opportunity when a nuclear weapon will be selected. SAM firing policy six will be called only when the input firing policy of the ship is six and there are conventional SAMs available.

SAMFP5

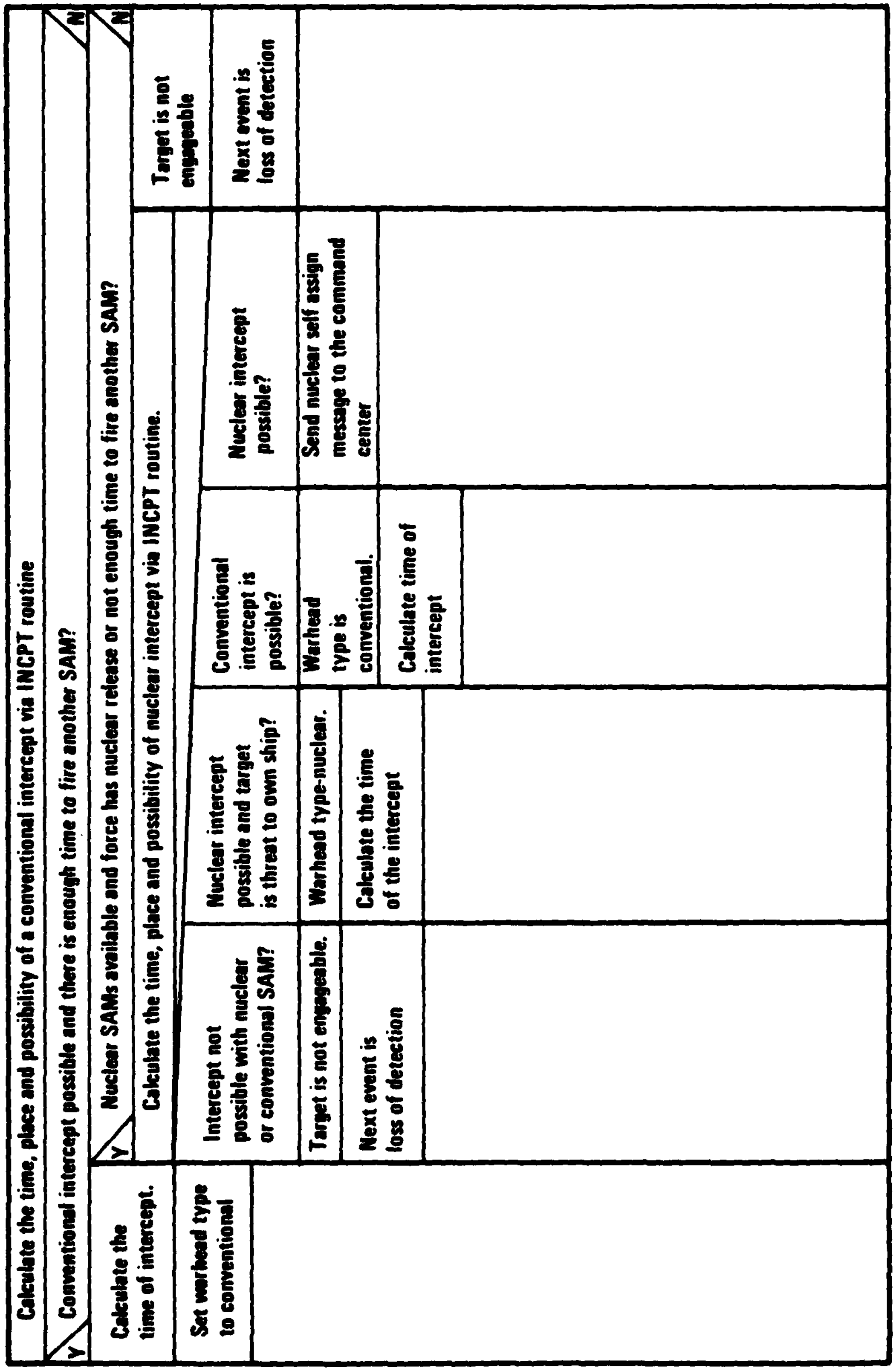


Figure 9-14 Flow Diagram for Subroutine SAMFP5

The possibility of a conventional intercept is calculated via the INCPT routine. If a conventional intercept is possible, the time of the intercept is calculated and the warhead type is set to indicate a conventional SAM. A two-missile salvo will be made if there are two conventional SAMs available.

When a conventional intercept is not possible, the possibility of a nuclear intercept is calculated provided there are nuclear SAMs available and the force has nuclear release. If the nuclear intercept is possible and the target is priority one, the warhead type will be marked nuclear and the time of the intercept calculated. When the target is not a priority one, a message is sent to the command center asking for permission to engage the target with a nuclear SAM. The SAM ship will wait for a response from the command center whether to engage the target with nuclear SAMs.

When a conventional and nuclear intercept is not possible, the warhead type will be set to zero, the SAM state will indicate that the target is not engageable, and the next event will be a loss of detection which means that no further processing of this target will be made by this ship.

Firing policies six and two differ in two respects in that six will:

1. fire a conventional two-missile salvo when it is not the last firing opportunity; and,

2. fire a nuclear SAM when it is the last firing opportunity.

The complete logical flow is shown in Figure 9-15, Flow Diagram for Subroutine SAMFP6.

9.3.3.7 Subroutine SAMFP7

Subroutine SAMFP7 contains the logic to support SAM firing policy seven;

S - L - S - L - S - L - S(n) - L - S(n) - L ...

SAM firing policy seven employs single conventionally armed missiles while the intercept is predicted to be outside the nuclear keep out range. The firing policy is changed to nuclear when the target comes within the nuclear keep out range of any ship in the force. SAM firing policy seven will be called only when the input firing policy of the ship is seven and there are conventional SAMs available.

The possibility of a conventional intercept is calculated via the INCPT routine. If nuclear SAMs are available and the force has nuclear release and the conventional intercept is possible, the shortest distance from the conventional intercept point to any ship in the force is calculated. If the shortest distance is less than the nuclear keep out range or the target is priority one and a

SAMFP6

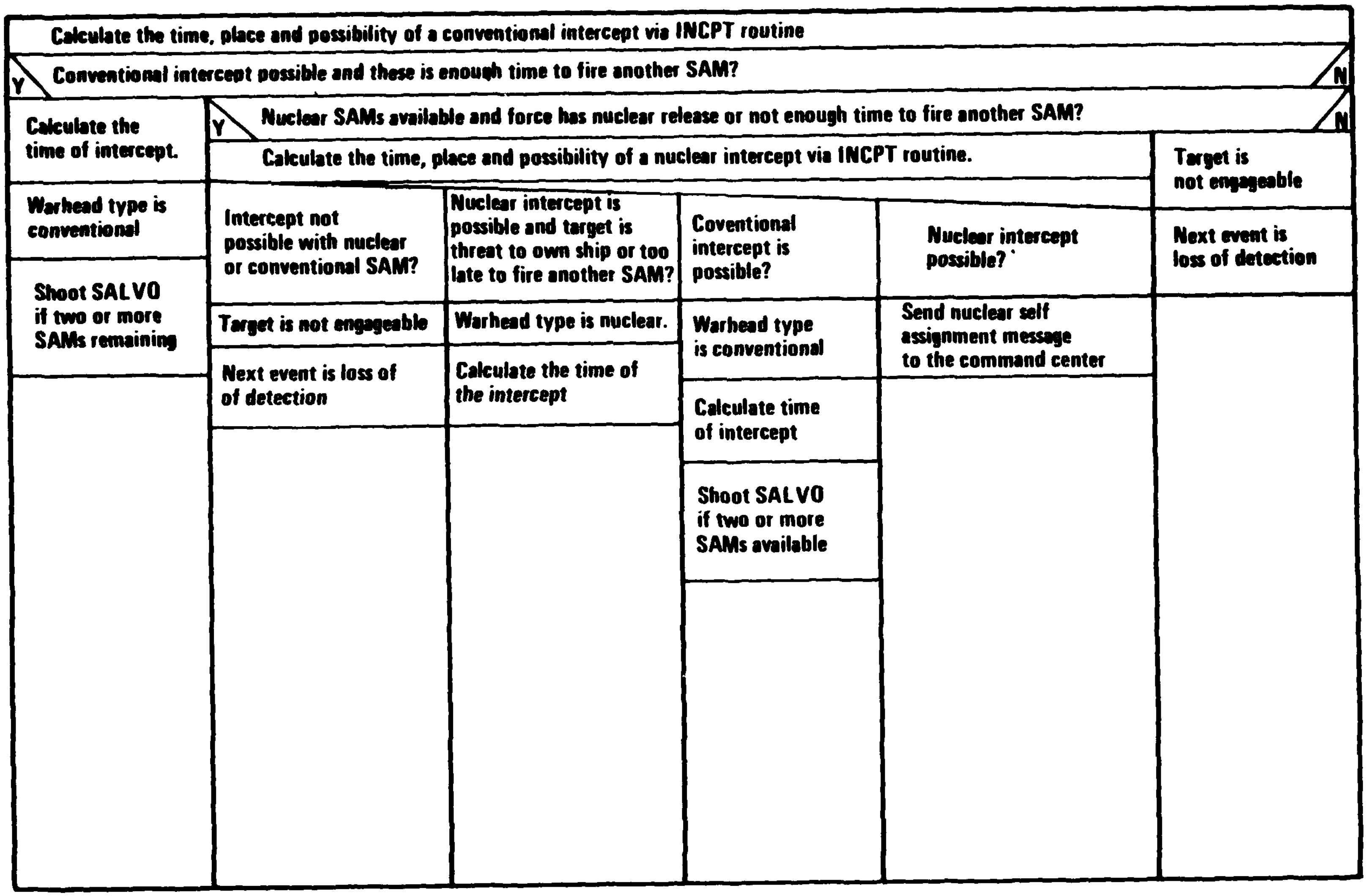


Figure 9-15 Flow Diagram for Subroutine SAMFP6

conventional intercept is not possible, the possibility of a nuclear intercept is calculated. If a nuclear SAM can intercept the target, the warhead type is set to indicate a nuclear SAM and the time of the nuclear intercept is calculated.

If nuclear SAMs couldn't be employed and a conventional intercept is possible, the warhead type is set to indicate a conventional SAM and the time of the conventional intercept is calculated.

When a conventional and a nuclear intercept is not possible, the warhead type will be set to zero. The SAM state will indicate that the target is not engageable, and the next event will be a loss of detection which means that no further processing of this target will be made by this ship. The complete logical flow is shown in Figure 9-16, Flow Diagram for Subroutine SAMFP7.

9.3.3.8 Subroutine SAMFP8

Subroutine SAMFP8 contains the logic to support SAM firing policy eight;

SS - L - SS - L - SS - L - S(n) - L - S(n) - L ...

SAM firing policy eight employs salvos of two conventionally armed missiles while the intercept is predicted to be outside the nuclear keep out range. The firing policy is changed to nuclear when the target comes within the nuclear keep out range of any ship in the force. SAM firing policy eight will be called only when the input firing policy of the ship is eight and there are conventional SAMs available.

The possibility of a conventional intercept is calculated via the INCPT routine. If nuclear SAMs are available and the force has nuclear release and the conventional intercept is possible, the shortest distance from the conventional intercept point to any ship in the force is calculated. If the shortest distance is less than the nuclear keep out range or the target is priority one and a conventional intercept is not possible, the possibility of a nuclear intercept is calculated. If a nuclear SAM can intercept the target, the warhead type is set to indicate a nuclear SAM and the time of the nuclear intercept is calculated.

If nuclear SAMs couldn't be employed and a conventional intercept is possible, the warhead type is set to indicate a conventional SAM and the time of the conventional intercept is calculated.

When a conventional and a nuclear intercept is not possible, the warhead type will be set to zero. The SAM state will indicate that the target is not engageable, and the next event will be a loss of detection which means that no further processing of this target will be made by this ship.

SAMFP7

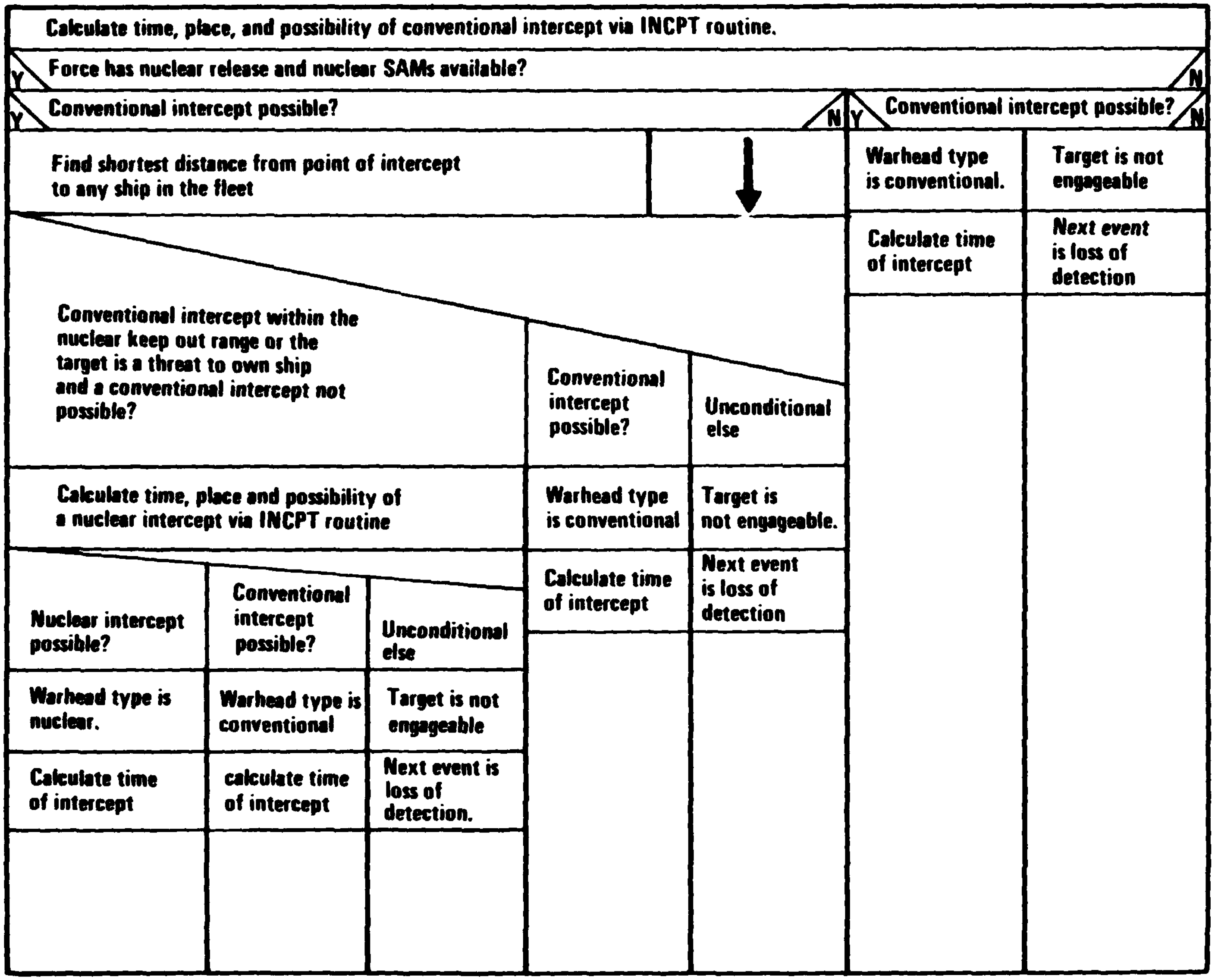


Figure 9-16 Flow Diagram for Subroutine SAMFP7

Firing policies eight and seven differ only in that eight will fire a conventional two-missile salvo at targets whose conventional intercept will be beyond the nuclear keep out range. The complete logical flow is shown in Figure 9-17, Flow Diagram for Subroutine SAMFP8.

9.3.4 Subroutine SAMGT3

Subroutine SAMGT3 contains the logic mid-course guidance with semi-active terminal guidance, a guidance channel to direct the illuminator for terminal homing must be scheduled prior to the calculated time of intercept. Since time has advanced to the point of being ready to launch, it is possible to compute the time it is desired to have the guidance channel available, as follows:

$$TDGAV = TNOW + TINTC - TILL$$

Where TDGAV is when the guidance channel is needed, TNOW is the time now, TINTC is the time of the intercept and TILL is the tie-up time before intercept for this guidance channel type.

The time a target will be illuminated will need to be greater for targets at long range than for targets at short range. In order to determine if a long or short illuminator time is required, the intercept range is tested against the illuminator crossover range. If the intercept range is less, the shorter illuminator time is used. If the intercept range is greater than the crossover range, then the longer illuminator time must be used and the time the illuminator is desired is recalculated as well as the subsequent tests. Crossover range and tie-up time are found in the fire control characteristics common (BFCHAR). The complete logical flow is shown in Figure 9-18, Flow Diagram for Subroutine SAMGT3.

9.3.5 Subroutine SAMLCH

The logic that controls the launch, intercept, and evaluation are determined in the routine SAM launch.

With guidance scheme 3, the SAM guidance type 3 (SAMGT3) routine must be called to schedule the illuminator or guidance channel.

If the guidance channel is available at or before the desired time, it may be possible to launch, given the intercept will occur before the target leaves the envelope.

In situations where the guidance channel is not available at the earliest desired time, then the launch time is determined by the time of guidance channel availability. SAM launch will be rescheduled for that time and the routine will be called again.

SAMFP8

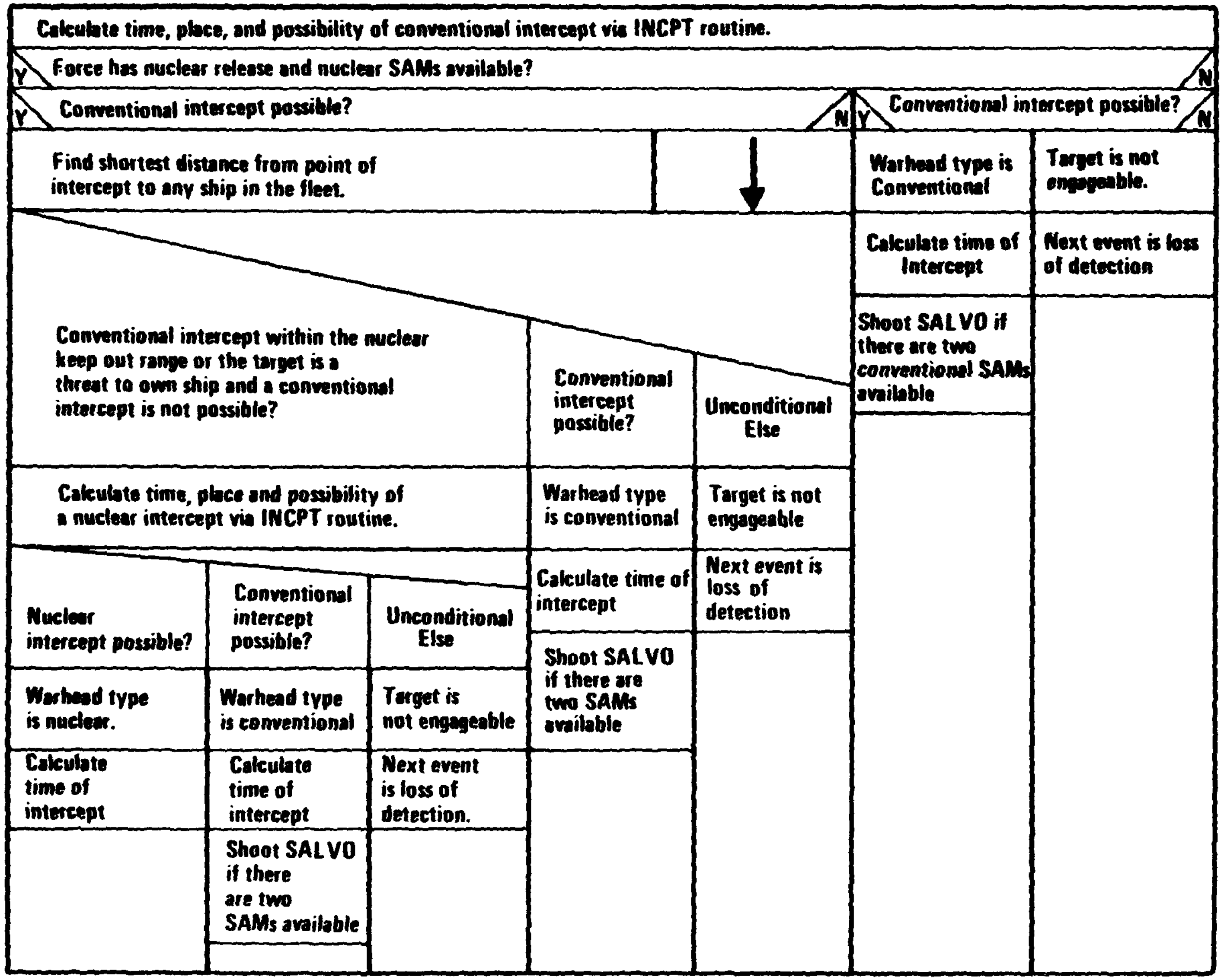


Figure 9-17 Flow Diagram for Subroutine SAMFP8

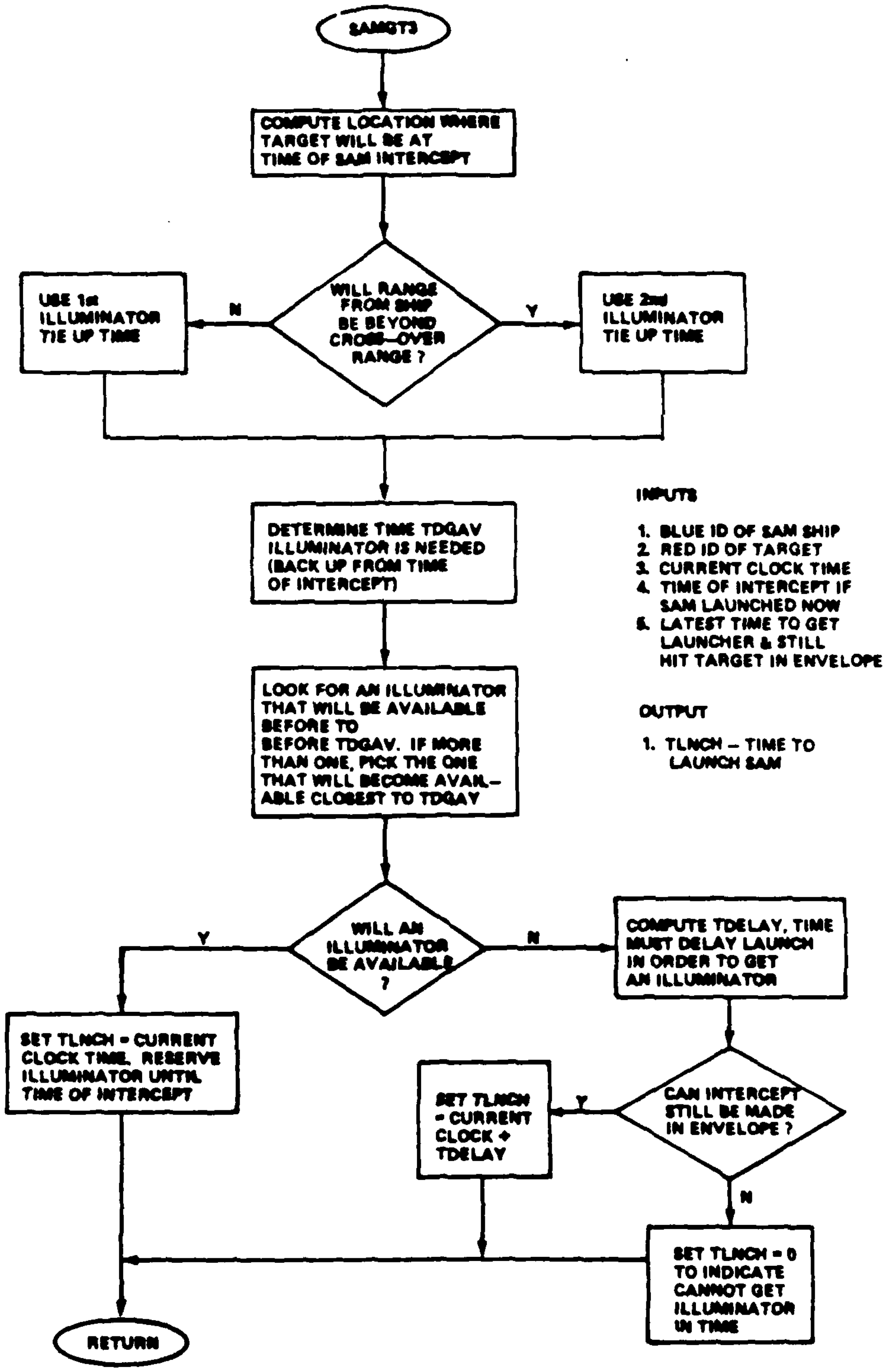


Figure 9-18 Flow Diagram for Subroutine SAMGT3

The SAM characteristics are summarized in Figure 9-19, SAM Characteristics. The first four parameters are needed for each SAM type. In addition, certain parameters are needed for each target type to be faced by the SAM. The five envelope coefficients are used to define the SAM intercept boundaries for each target type. The probability of hit table defines three hit probabilities for the three corresponding SAM times of flight. If the intercept occurs during a time of target acceleration from level flight to dive on target (in the gap), then a fourth probability of kill is used.

An independant Bernoulli trial is used to predict the outcome of the SAM intercept. The parameter of the distribution is a complex one initially based on table lookup. Three probabilities of kill for three flight times are used. For a two-missile salvo, the probability of kill is treated as the combination of two independent trials. For diving missile targets, provision has been made for reduced missile performance during the transition from level flight to dive. The time of intercept is compared to the time of missile dive and a lower probability of hit is used if the intercept will occur during a time of target acceleration; i.e., in the gap.

If a hit is predicted and there are no intervening events, the result is a hit. The time of hit is set to the intercept time for the SAM making the hit. The target is removed from the simulation at the time of the intercept, and the fire control channel is released after an additional delay for the evaluation of the intercept.

In the event of a miss, nothing happens at the time of the intercept. The next event will be at the end of the evaluation period and since the fire control channel is already locked on the target, only an appropriately loaded launcher is needed to fire again. The complete logical flow is shown in Figure 9-20, Flow Diagram for Subroutine SAMLCH.

9.3.6 Subroutine SMINCP

Subroutine SMINCP determines engageability and establishes the priority for targets detected by the ship. Surface-to-Air Missiles must be on the ship for processing to proceed.

Next the closest point of approach (CPA) to own ship is computed. This and some of the parameters to be discussed are shown in Figure 9-21, SAM Ship Decision Geometry. Targets whose CPA does not come within the cross range of the SAMs available are not considered further.

Targets that will come within a critically close range (HOTCPA) are a threat to own ship and are labeled a priority one or five target. A self-assign notice is sent to inform the command center of the intent to fire.

- MAXIMUM RANGE (N.MI.)
- MINIMUM RANGE (N.MI.) (SM3)
- NOMINAL SPEED (KNOTS)
- SAMTYP – 1,2, OR 3

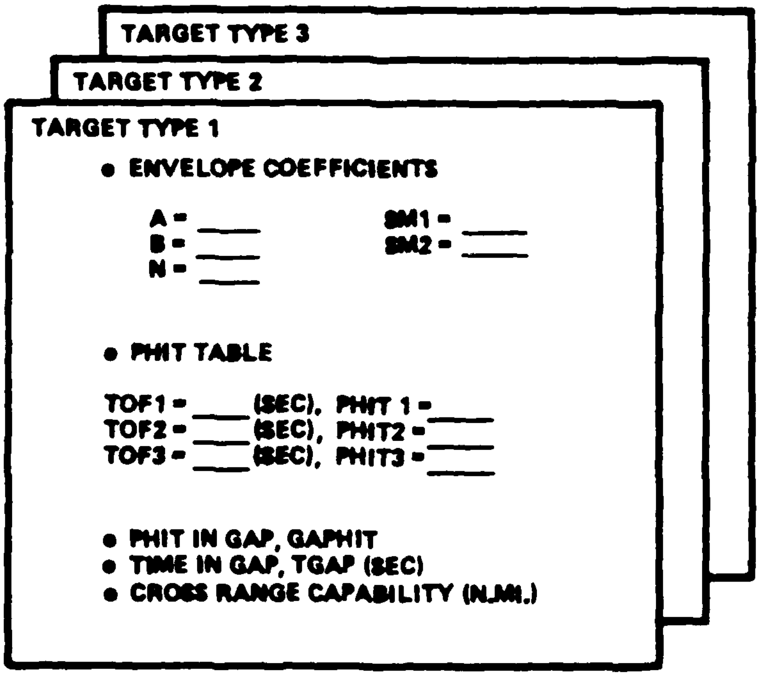


Figure 9-19. SAM Characteristics

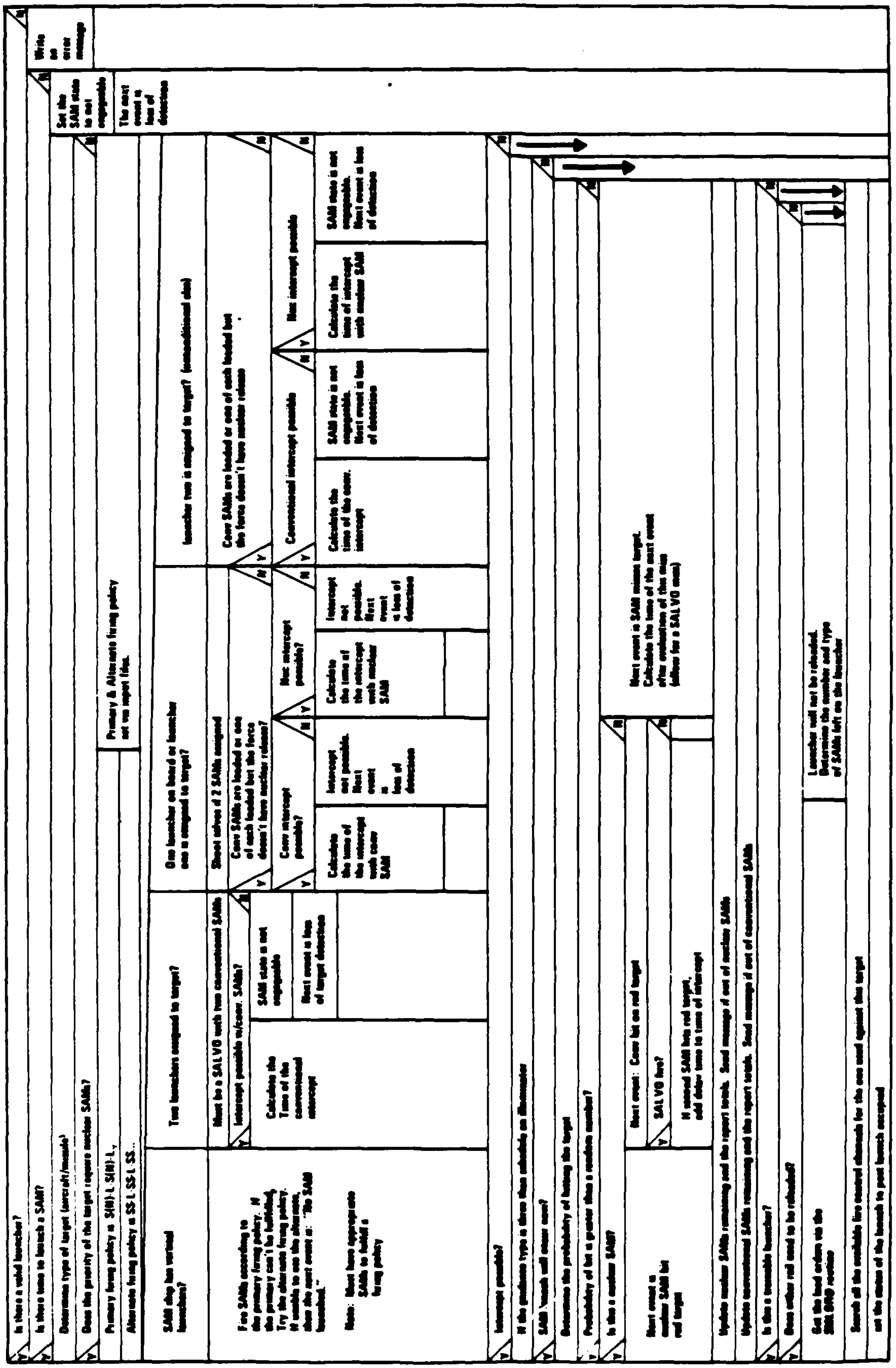


Figure 9-20 Flow Diagram for Subroutine SAMLCH

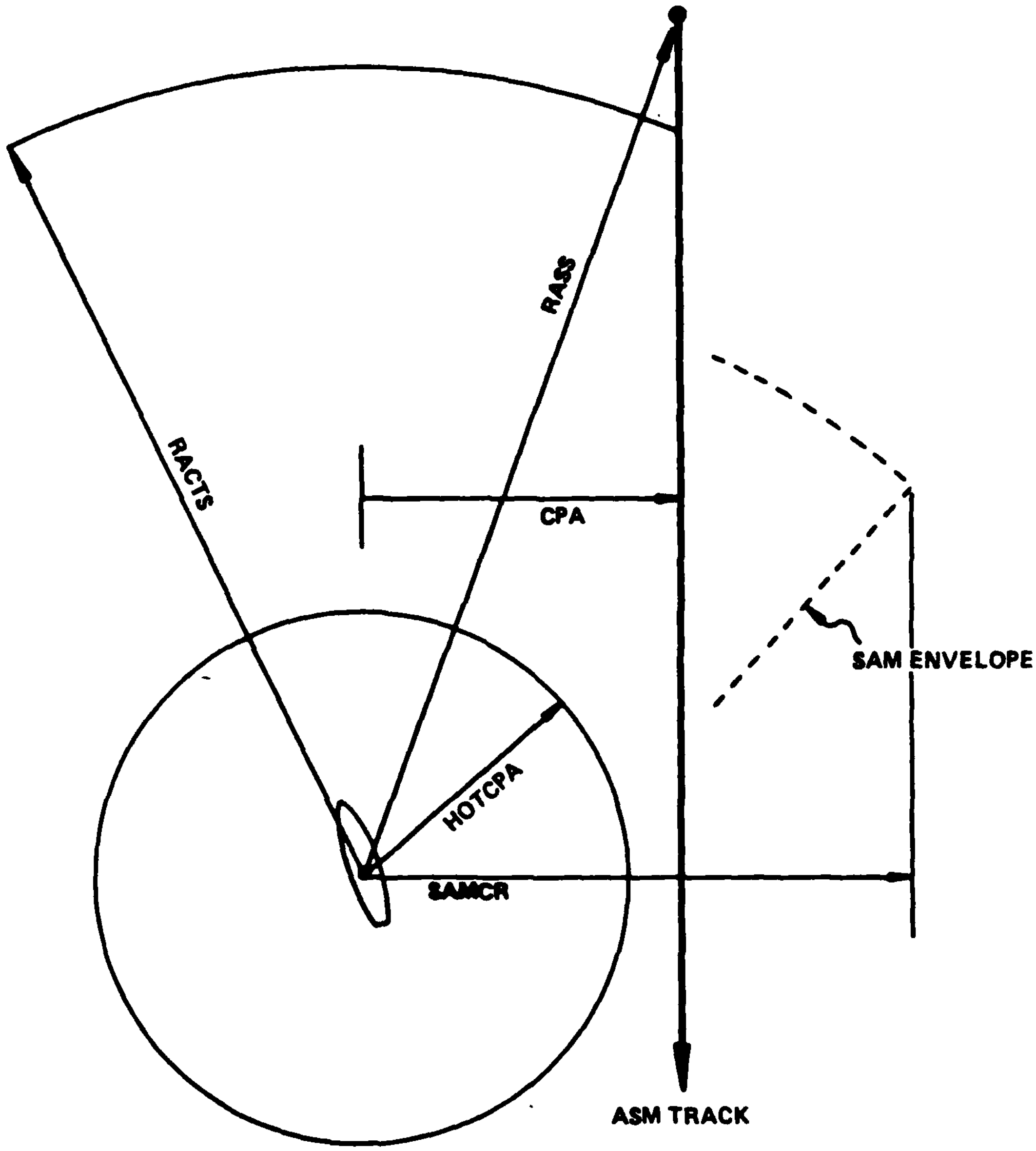


Figure 9-21 SAM Ship Decision Geometry

Targets that do not threaten own ship will be assigned priority three status (self assignment) unless and until the command center assigns the target to the ship. If the ship is using coordination doctrines one or two, it will check to see if the target will be in its sector before a self assignment.

After the priorities are set, the times the target enters and exits the SAM envelope are determined (see Figure 9-22, SAM envelope). From these times, the earliest time the fire control is needed is computed. If current time is already past that point, then it must be determined if the intercept can be made before it leaves the envelope. In the first case, the time at which a fire control channel should be seized (see Figure 9-2, Functional Flow with SAMTYP 1 or 2) is established. In the second case, if the intercept can be made, then the fire control channel should be seized as soon as possible. When an intercept is not possible because the target is past the SAM envelope or beyond its cross range capability, the target is marked as not engageable by this ship. No further processing is done unless the target changes course.

Figure 9-23, Find Fire Control Channel, show the sequence for finding a fire control channel. First choice is an unoccupied one. With priority one targets, an occupied channel may be preempted if no unoccupied ones are available.

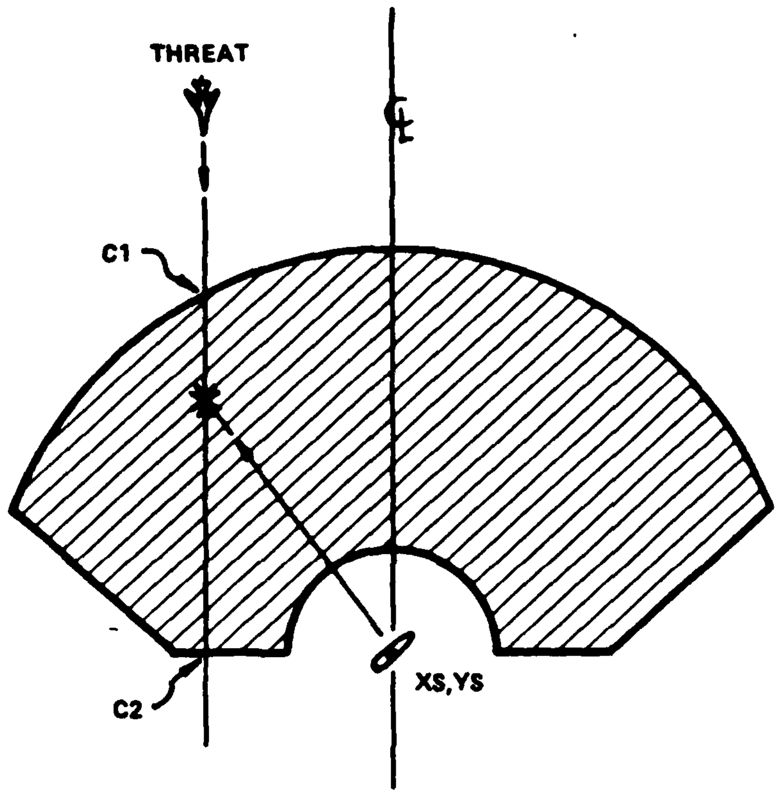
Following the flow shown in Figure 9-4, SAM Ship Events, the next event for the ship will occur when a fire control channel becomes available. The subroutine calculates the last time that a launcher is needed in order to engage the target using fire control lock-on and track settling times if required. The engageability of the target is determined by the launcher requirement. The complete logical flow is shown in Figure 9-24, Flow Diagram for Subroutine SMINCP.

9.3.7 Subroutine SMLORD

Subroutine SMLORD determines what type SAM, nuclear or conventional, and the number of missiles to load next on an empty launcher and schedules the return to ready status for the launcher. A load order is created which is sent to the GPSS. Note that the launcher is not loaded at this point. The actual loading will occur in SAM load (SMLOAD) which is called by the GPSS module.

Orders are based on the firing policy of the ship, the number of empty rails, number of launchers to be loaded, and the future intercepts possible for the target against which the ship has just launched.

SAM load order is called from SAM launch and is used only for trainable launchers since vertical launchers never need to be loaded. The load orders will be stored in the SAM ship launch state array. The complete logical flow is shown in Figure 9-25, Flow Diagram for Subroutine SMLORD.



Threat crosses SAM envelope at TC1 and TC2, solution of straight line (target track) and truncated superellipse (SAM envelope)

Figure 9-22 SAM Envelope

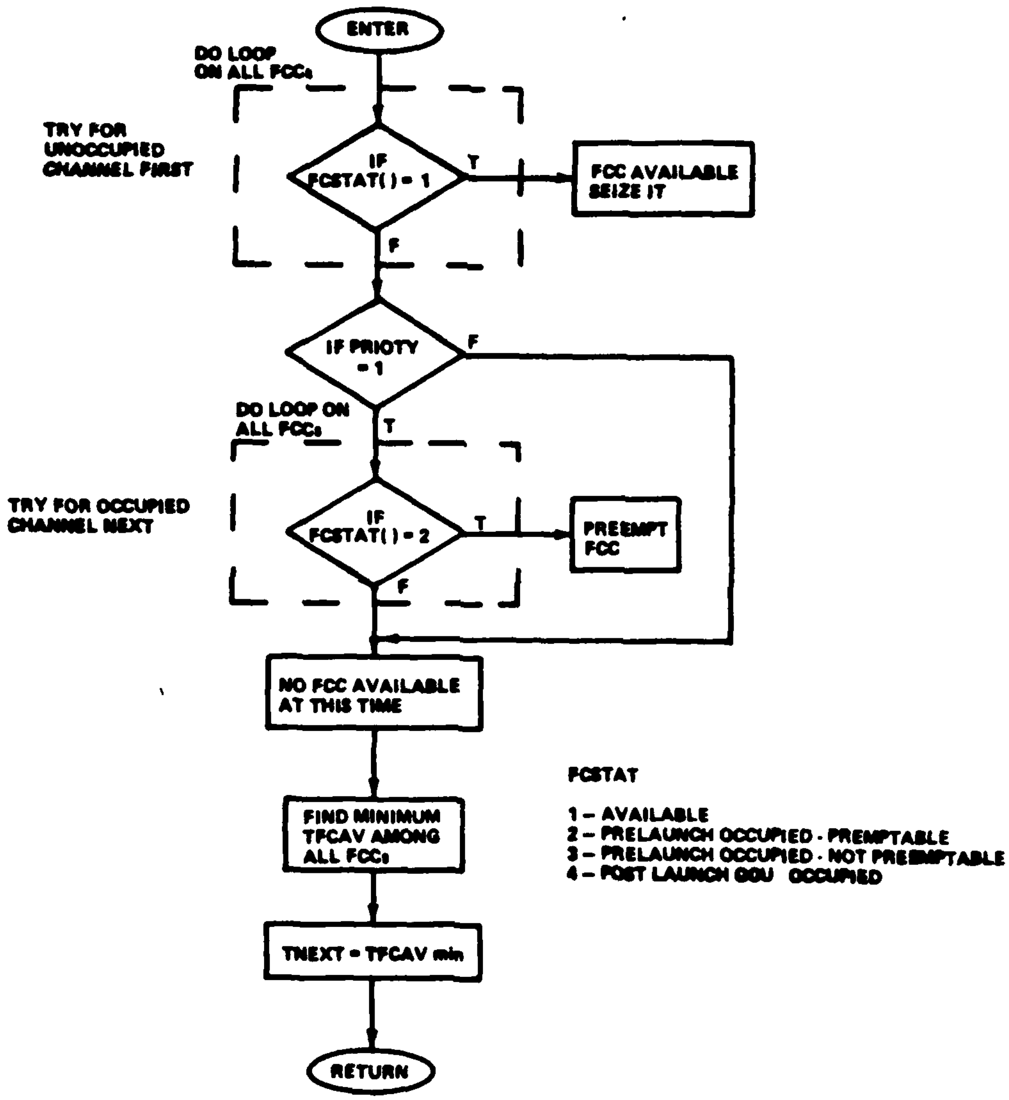


Figure 9-23 Find Fire Control Channel

SMINCP

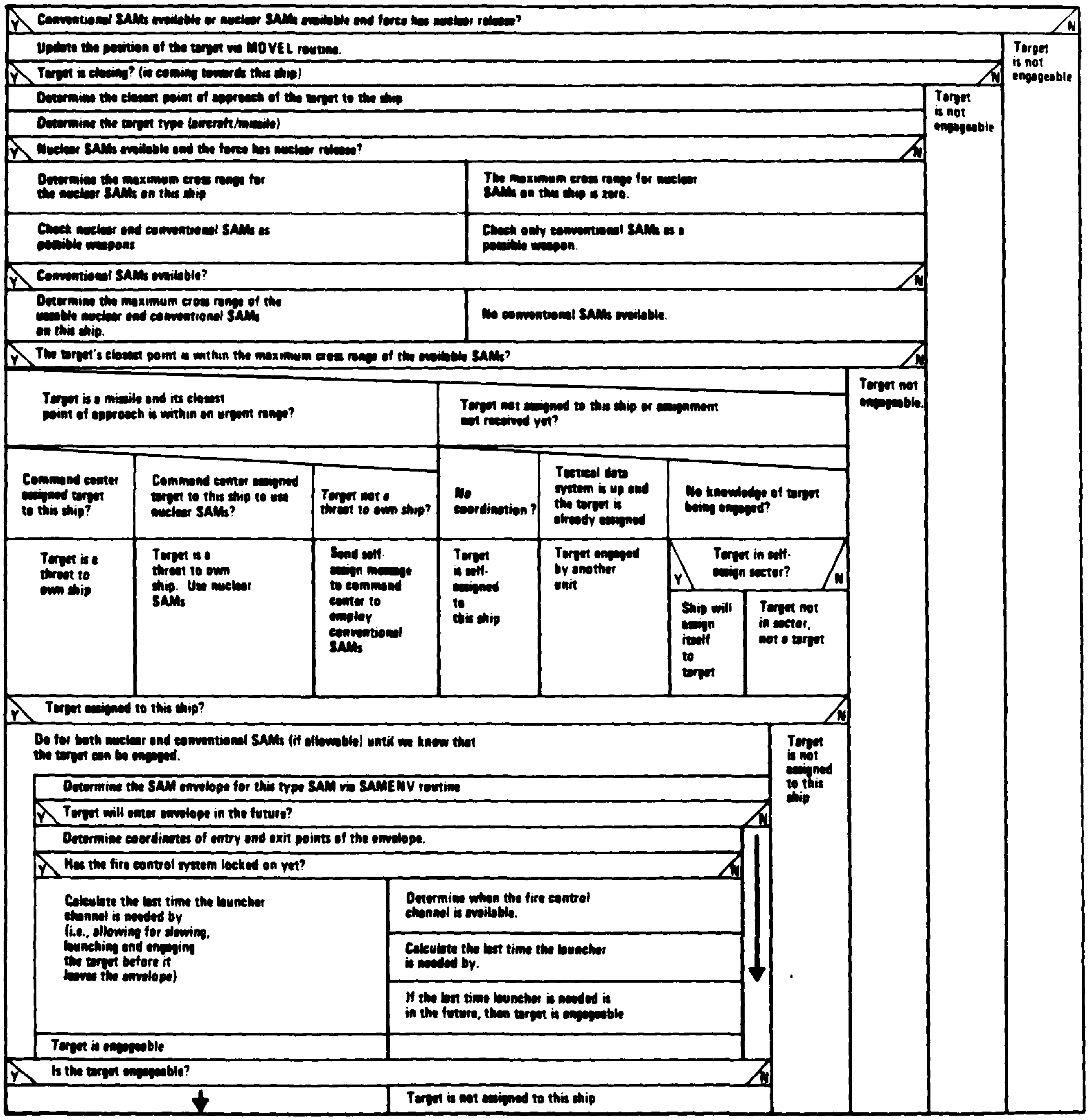


Figure 9-24 Flow Diagram for Subroutine SMINCP

SMLORD

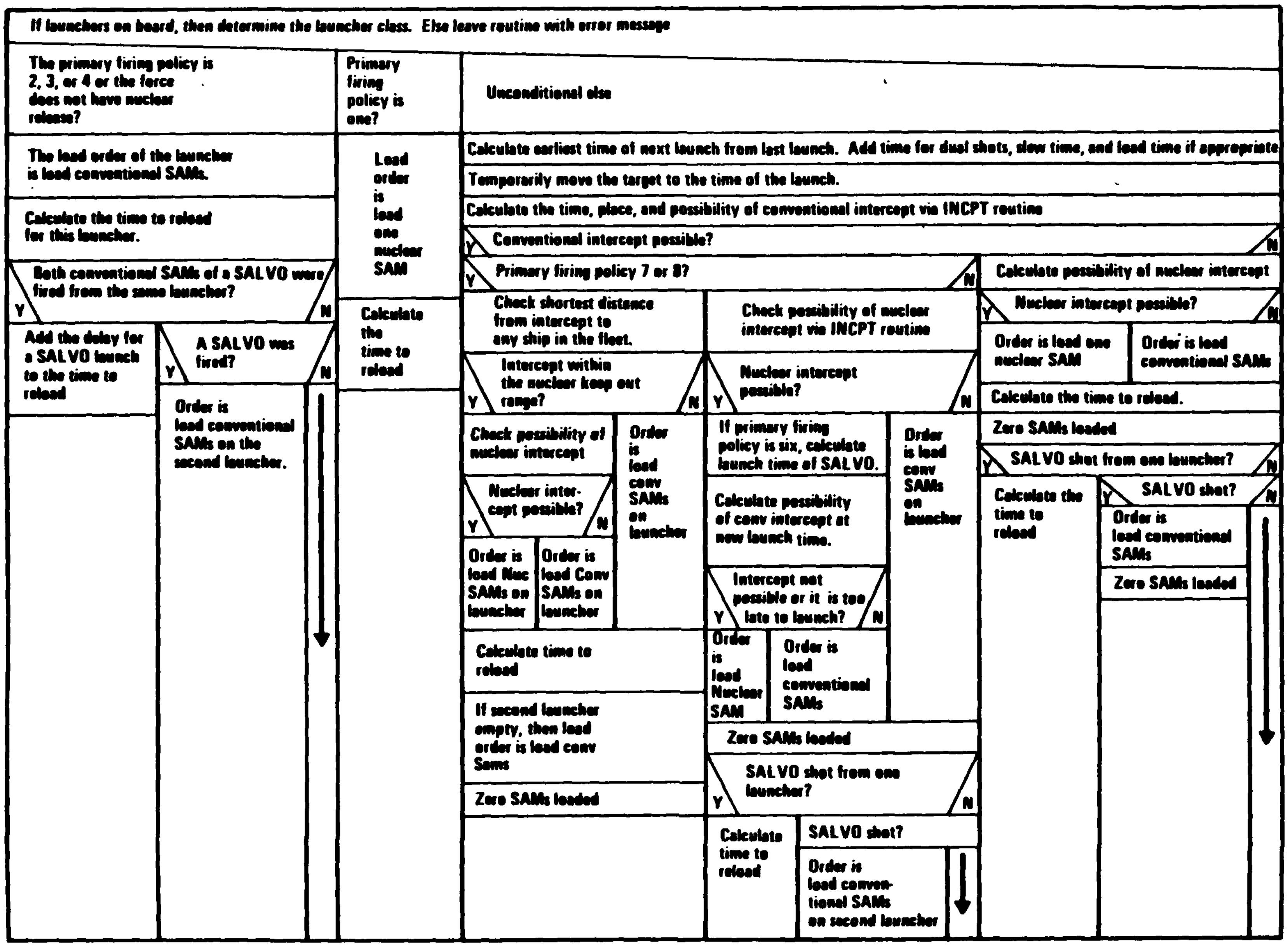


Figure 9-25 Flow Diagram for Subroutine SMLORD

9.3.8 Subroutine SMLOAD

Subroutine SMLOAD loads missiles on launchers based on the load orders received from Subroutine SMLORD, the number of surface-to-air missiles in the ship's magazines and the launcher configuration of the ship. Subroutine SMLOAD is used only for trainable launchers since vertical launchers never need to be loaded. The complete logical flow is shown in Figure 9-26, Flow Diagram for Subroutine SMLOAD.

9.3.9 Subroutine SMLSEL

Subroutine SMLSEL contains the logic used to select missile launchers with which to engage a target from the launchers loaded on the ship and not assigned to any other target. The type assignment as reflected in the target priority, the SAMs loaded, and the primary and alternate firing policies for this ship are used in the logic. If a launcher cannot be assigned at this time, the target is placed on a list to wait for a launcher to become available. Subroutine SMLSEL is used only for ships with trainable launchers since with vertical launchers each SAM is in a launcher. The complete logical flow is shown in Figure 9-27, Flow Diagram for Subroutine SMLSEL.

9.3.10 Function TBRG

Function TBRG returns the true bearing from and observer to a point that is X-units East and Y-units North of the observer's position. Result is in radians in the range zero to two pi. Returns zero for inputs of (0.0, 0.0). The complete logical flow is shown in Figure 9-28, Flow Diagram for Function TBRG.

SMLOAD

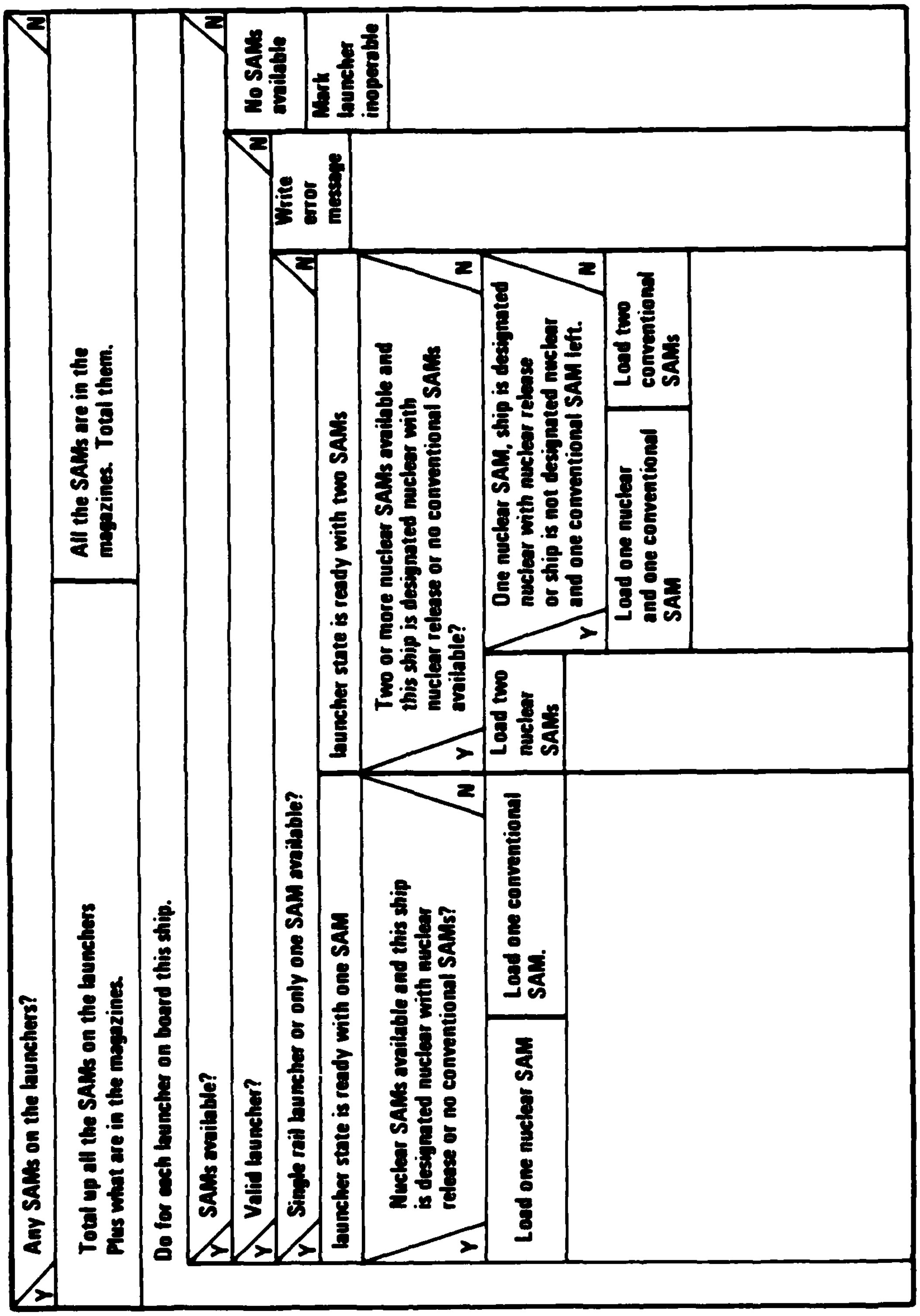


Figure 9-26. Flow Diagram for Subroutine SMLORD

SMLSEL

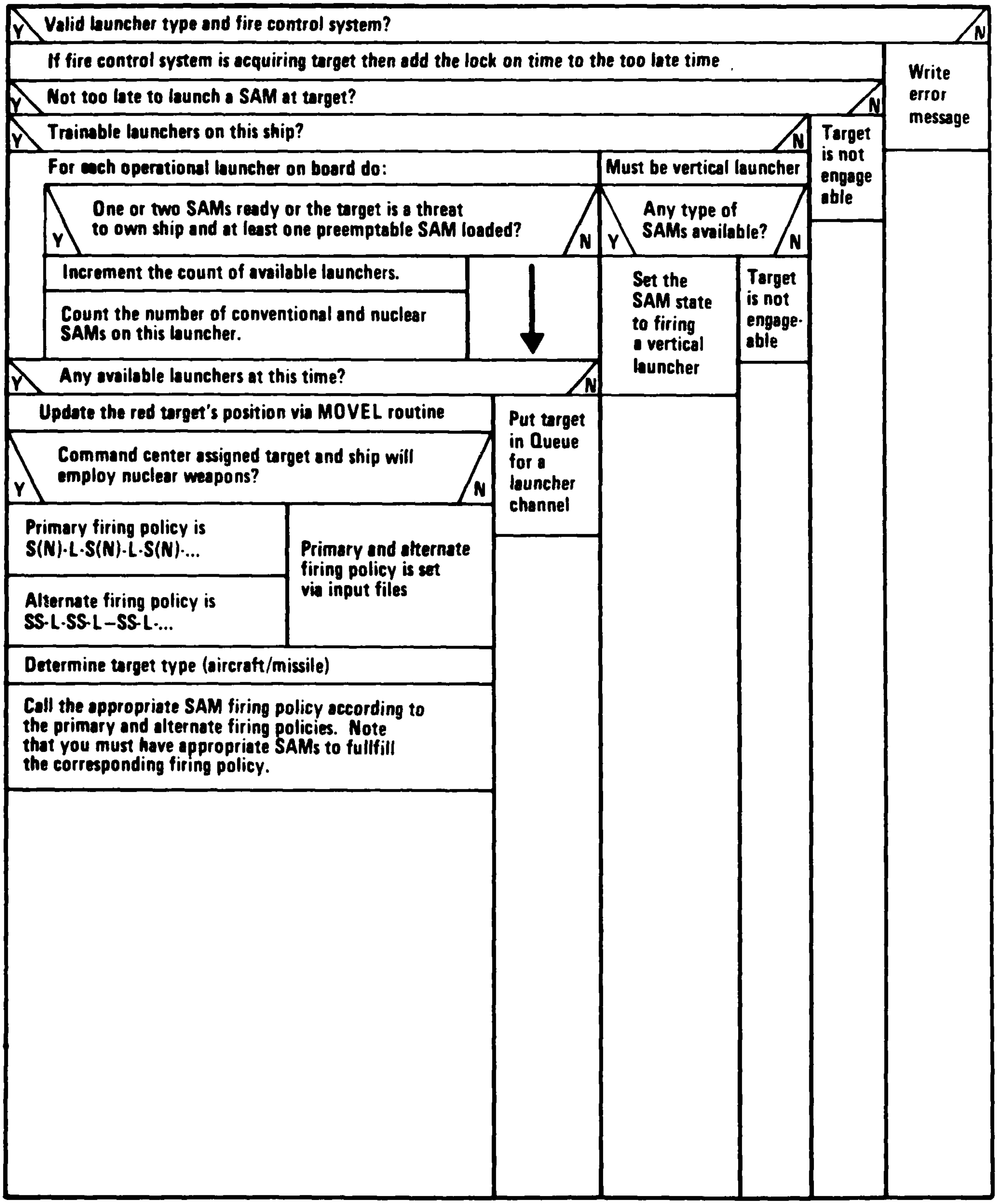


Figure 9-27. Flow Diagram for Subroutine SMLSEL

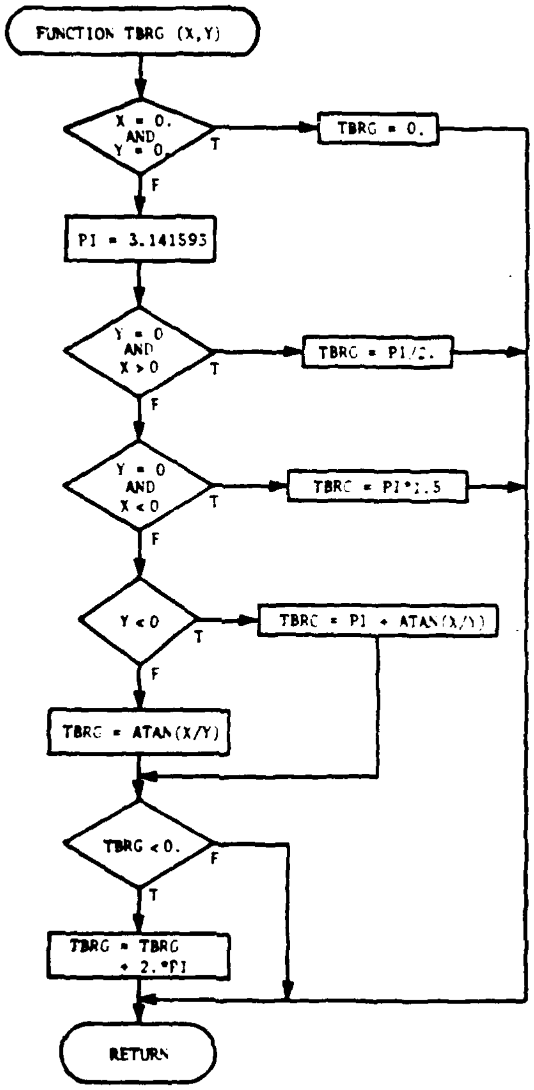


Figure 9-28. Flow Diagram for Function TBRG

CHAPTER 11

TERMINAL DEFENSE AND DAMAGEASSESSMENT MODULE

Blue damage is assessed during a simulation run so that the attrition of both Red and Blue capabilities can be dynamically accounted for and used to modify the ensuing course of events. Damage that the fighters inflict on Red aircraft and Red missiles is computed in the interceptor module. Damage that SAMs inflict on their intended targets is computed in the SAM Ship Module. Damage caused to embarked aircraft on aircraft carriers is assessed within the aircraft carrier module.

The Terminal Defense and Damage Assessment module computes the damage that Red missiles inflict on the targeted ships when all defenses are penetrated, and it computes the nuclear weapon damage to all Red units and all Blue units, regardless of whether the nuclear weapon was a Red missile, Blue SAM or Blue AIM, and regardless of whether the burst was on the intended target or elsewhere as a consequence of defensive action.

The Terminal Defense and Dammage Assessment module handles conventional weapons and nuclear weapons separately as described in Sections 11.2 and 11.3, respectively.

11.1 TERMINAL DEFENSE

There are two general types of terminal defense systems that may exist on a ship. Active systems, such as missiles and guns, produce "hard" kills. Passive systems, such as deceptive electronic countermeasures and radio frequency decoys, produce "soft" kills. Any given ship may use one or more active or passive systems which makes it difficult to model the entire Point Defense system as an aggregate system.

Some of the factors which affect the performance of each system are:

- o Attack density;
- o Reaction time;
- o Kill distance from the ship;
- o Fratricide;
- o Jamming;
- o Missile or ordnance supply;
- o Weather; and,
- o Search radar and sensor integrations.

All of these factors are wrapped up in a sequence of stochastic outcomes depending on the specific mix of systems within the combined Point Defense as well as the specific ASCM type. Figure 11-1, Flow Diagram for PNTDEF illustrates the general implementation of the Point Defense model. PNTDEF (PoiNT DEFense) uses the PARAM, BLUNIT, PROGRM, RDUNIT, REPRT, RMCHAR, RSM, SHCHAR, and SHSTAT commons. Subroutine GAUSS is called to generate normally distributed random numbers and function XYDIST is used to calculate the horizonal distance between two points.

The number of ASCM's penetrating to each ship is scored by ASCM type. ASCM type is available as RUMTY in the RDUNIT common. ASCM type is used to identify nuclear ASCMs and to determine Point Defense outcome. Various outcomes are possible for Nuclear armed ASCM's defeated by the point defenses:

- o Destruction of airframe and warhead;
- o Destruction of airframe only (warhead detonates);
- o Warhead detonated (salvage fusing);
- o Missile distracted but warhead functions

Damage assessment is done whenever a detonation results. ASCMs with conventional warheads are either defeated or passed to damage assessment for evaluation.

Damage by conventional ASCMs is evaluated as described in Section 11.2. The nuclear ASCM damage is evaluated to a greater level of detail described in section 11.3

PNTDEF

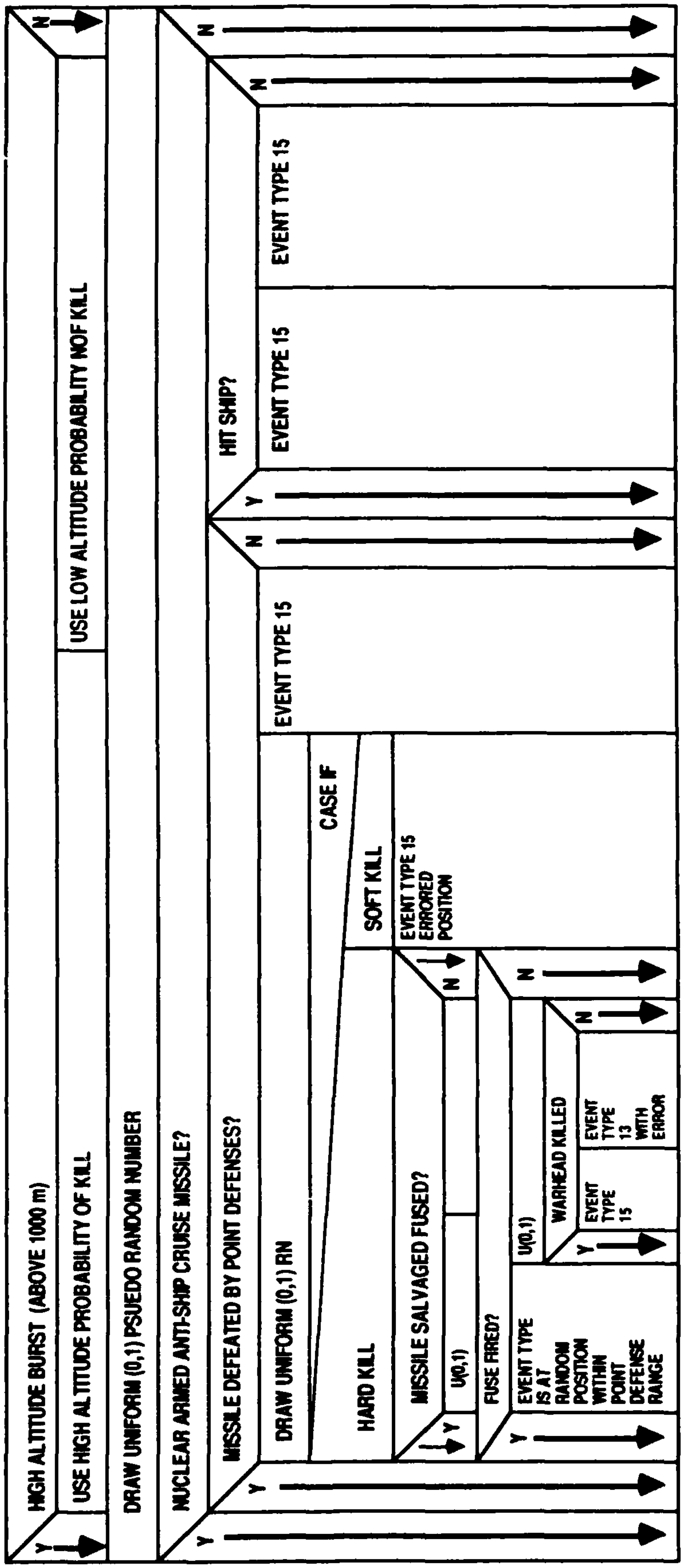


Figure 11-1 Flow Diagram for PNTDEF

11.2 CONVENTIONAL WEAPONS

Each non-nuclear ASCM that reaches a ship will be evaluated to determine the damage that will be caused to the ship. Only those systems simulated in NADS are examined. The design is intended to be simple and yet accommodate multiple ship configurations. The functions that may be damaged are:

- o aircraft launching (carriers only);
- o detection capability;
- o SAM launch capability;
- o Point Defenses

The general scheme is to have three possible operating capabilities each for aircraft launch, SAM launch, and point defense with two possible operating capabilities for detection capability. The aircraft carrier may have normal aircraft launching capability with four catapults available, reduced capability with two catapults available, or none available. These availabilities effect the rate-of-launch of aircraft in the CV Operations module. The SAM launching capability is similarly conceived except the number of launchers may differ from ship to ship. The launcher acts as a proxy for the entire SAM system. A fully "up" system on ships that have either two launchers or a vertical launching system (VLS) is the normal initial state. The reduced capability leaves the ship with one launcher or one-half of the VLS cells. A ship with a single launcher starts in this state. The third level is no SAM launching capability. Point defenses capability is reduced by modifying the probability of kill associated with the system. Otherwise, the mix of outcomes is not altered. When the ship reaches fifty percent weapon impairment, based on the number of hits, the kill probabilities are reduced to half the input value. At one hundred percent impairment, the point defenses no longer have capability. The detection capability is either normal (fully "up") or none.

A system is changed from one capability to another after the number of penetrators exceeds the value input by the user specifying how many hits are required to cause impairment. Penetrators are simply counted until they exceed the input threshold, then the system capability is reduced. Because the SAM system is so dependent on the detection system, loss of the detection system automatically reduces the SAM capability by one level.

Figure 11-2, Flow Diagram for DANONN, illustrates the general implementation for conventional ship damange. DANONN (DAmage NON-Nuclear) uses the PARAM, BLUNIT, CCSTAT, MESSG, PROGRM, RDUNIT, REPRT, SHCHAR, and SHSTAT commons. DANONN calls the subroutine MKMSG to send status reports to the command center.

DANONN

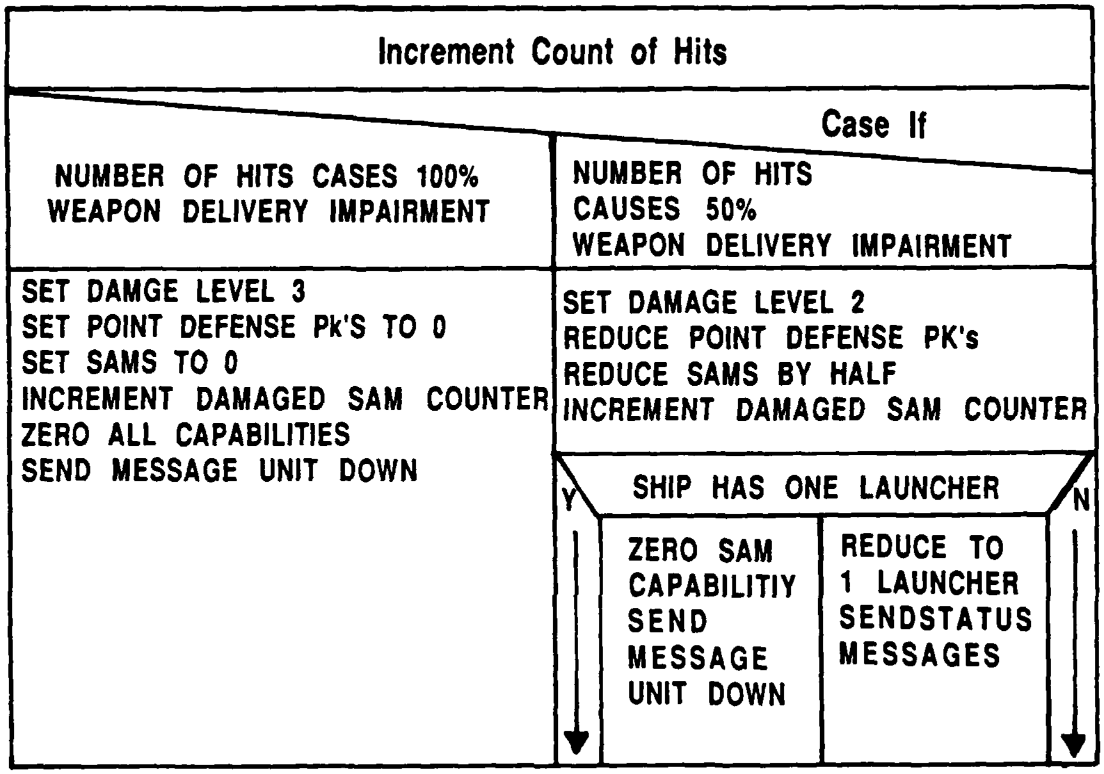


FIGURE 11-2 FLOW DIAGRAM FOR DANONN

11.3 NUCLEAR WEAPONS

Whenever a Red or Blue nuclear warhead burst occurs, subroutine NUCLER is called to ascertain the consequent damage to all Red and Blue units. Because NUCLER is a lengthy routine (it calls 20 other subroutine and function subprograms), a significant amount of code is devoted to editing out the cases that do not need to be computed.

NUCLER uses a number of FORTRAN common blocks to obtain most of its input data and to store most of its output data. Its specific calling arguments are: a Blue ID number (BUID), a Red ID number (RUID), a flag indicating whether the Red ID is the bursting warhead or the target (MFLAG), the time of the burst which is the current clock time (TB), list of Blue and Red units suffering higher levels of damage from this burst (NEWBID and NEWRID) and the number of those units (NOBLU and NORED).

It is assumed that there are five basic circumstances in which NUCLER will be called:

- CASE 1: Red nuclear missile penetrated all defenses and burst at its intended X-Y coordinates and design height of burst. The targeted ship is presumed sunk if within the fireball region.

- CASE 2: Red missile suffered previous airframe and/or guidance system damage, but its warhead and fuzing system is unimpaired. It bursts at the design height of burst but at an erroneous X-Y position.

- CASE 3: Red missile was salvage fuzed, which could occur at a point on its flight path if it had salvage fuzing.

- CASE 4: Nuclear surface-to-air missile (SAM) was burst at the position of its intended target. The SAM ship module will already have determined the success or failure of its shot, and it is assumed here that it was successful or the warhead would not have been given a detonation command. Such an assumption is required because NADS does not maintain position data along SAM trajectories.

- CASE 5: Nuclear air intercept missile (AIM) was burst at the position of its intended target. The fighter (VF) interceptor module will have determined the success or failure of its shot, and it is assumed here that it was successful or the warhead would not have been given a detonation command. Such an assumption is necessary because NADS does not maintain position data along AIM trajectories.

In all cases, it is the X-Y-Z position of the Red ID that defines the position of the burst. The Blue ID is needed to determine the warhead characteristics for Cases 4 and 5. The Blue ID is the launching ship or fighter, which yields the unit's platform type number and the warhead type number (IWHB).

There are eight nuclear environments or weapon effects that are individually computed and compared with relevant threshold values for damage assessment. The eight environments are:

1. Blast Overpressure;

2. Blast Dynamic Pressure;

3. Overpressure Impulse;

4. Particle Velocity, Normal to Flight Path;

5. Particle Velocity, Parallel to Flight Path;

6. Prompt Gamma Peak Dose Rate;

7. Neutron Fluence; and,

8. Thermal Fluence.

The damage threshold values against which the computed environment levels are compared are stored in several FORTRAN common blocks that represent the characteristics of the various types of missiles and platforms in the game. The general form of the vulnerability or response arrays is

XXRSP(N,J,K),

where the XX indicates the particular block of characteristic data, e.g. RMRSP for data on Red missiles. The N is the index to a specific platform type, the J is the index to a particular nuclear environment, and K is a designator for the level of damage. The range of K varies among the different functional types. For ships, K goes up to six representing successively increasing levels of damage, viz:

Level 1: Loss of air search radar;
Level 2: 50% weapon delivery impairment;
Level 3: 100% weapon delivery impairment;
Level 4: 90% mobility impairment;
Level 5: 90% sea worthiness impairment; and,
Level 6: Ship destroyed (sunk).

For example, if SHRSP(4,1,2) = 8.5 then a ship whose class is indicated by 4, if exposed to blast overpressure (1), will suffer a 50% weapon delivery impairment

(2) if the overpressure exceeds 8.5 psi. Note that it also loses its radar. The model assumes that all levels of damage lower than the current level are incurred.

Red missiles are characterized by only three levels of damage, viz:

Level 1: Airframe or guidance damage that would neutralize a nonnuclear missile but not a nuclear missile;
Level 2: Fires the salvage fuzing (if any); and,
Level 3: Disables the warhead.

Aircraft have only two levels of damage, viz:

Level 1: Loss of mission capability; and,
Level 2: Loss of aircraft.

NUCLER's use of subroutines, function subprograms, and labelled common blocks is illustrated in Figure 11-3, Nuclear Weapons Subroutine and Function Calling Tree, Figure 11-4, Nuclear Weapons Subroutine and Function Relational Matrix, and Figure 11-5, Nuclear Weapons Subroutine and Function Labelled Common Usage. Also included are flow diagrams of the major subprograms in the nuclear module.

- NUCLER
  - SCALE
    - MATM62
  - NUCSLEV
    - TERPL2
  - XYZDST
  - TSHK
  - MULLER
    - BLINT
  - RSHK
  - BLAST
    - MATM62
    - XYDIST
    - PFA
    - MYSTEM
    - PMR
      - PFA
      - XYDIST
    - VPARTS
      - XYDIST
      - XYZDST
    - PULSE
      - TERPL2
  - RHOX
    - TERPL2
    - MATM62
  - FENV
    - TERPL

Figure 11-3 (U) Nuclear Weapons Subroutine and Function Calling Tree (U)

CALLED \ CALLER		SUBROUTINES								FUNCTIONS						
		BLAST	BLINT	FENV	MULLER	NUCLER	NUCSLEV	SCALE	VPARTS	PMR	PULSE	RHOX	TERPL	TERPL2	XYDIST	XYZDST
SUBROUTINES	BLAST					●										
	BLINT				●											
	FENV					●										
	MATM62	●						●				●				
	MULLER					●										
	NUCSLEV					●										
	SCALE					●										
	VPARTS	●														
FUNCTIONS	I4TO2					●										
	MYSTEM	●														
	PFA	●								●						
	PMR	●														
	PULSE	●														
	RHOX					●										
	RSHK		●			●										
	TERPL			●								●				
	TERPL2						●				●					
	TSHK					●										
	XYDIST					●			●	●						
	XYZDST					●			●							

Figure 11-4 (U) Nuclear Weapons Subroutine and Function Relational Matrix (U)

		COMMON BLOCKS																					
		APSTAT	BLUNIT	BMCHAR	CONST	ENV	MISC	NUCLOG	NWCHAR	PARAM	RACHAR	RDUNIT	REPRT	RMCHAR	RSM	SCALT	SHCHAR	SMCHAR	VFSTAT	VFCHAR	VOCHAR	VWCHAR	VWSTAT
SUBROUTINES	NUCLER	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
	BLAST				●	●		●		●	●			●		●	●			●	●	●	
	BLINT					●				●						●							
	FENV				●	●	●	●	●	●													
	MULLER				●		●																
	NUCSLEV				●			●	●	●													
	SCALE															●							
	VPARTS					●	●			●													

Figure 11-5, Nuclear Weapons Subroutine and Function Labelled Common Usage

The general scheme of NUCLER, Figure 11-6, is to first establish the burst position, the type of warhead, the effective lifetime of the burst and store this burst data for post-simulation review. The effective lifetime of the burst (T1) is based on an empirical equation that approximates the data in the EM-1 manual for a restricted set of conditions, viz. that the burst altitude be less than 20 km for communication frequencies or less than about 30 km for radar frequencies. The approximation is:

$$T1 = Tb + \left(40 + 4.5 \left(.001 \times Zb \right)^{1/2} \right) Y^{.08} ,$$

where

Tb - burst time (sec),
Zb - height of burst (m), and
Y - yield (kt).

NUCLER

Initialize Parameters

Burst by Blue missile or, Red unit still in game?

Y:

Burst by Red missile?

Y: STORE RED PLATFORM TYPE, WARHEAD TYPE AND BURST COORDINATES

N: STORE BURST COORDINATES

BLUE SAM BURST?

Y: STORE SAM MISSILE TYPE, SHIP TYPE AND WARHEAD TYPE

N: STORE AIM MISSILE TYPE, AIRPLANE PLATFORM TYPE AND WARHEAD TYPE

Store burst data into NUCLOG common

Compute effective yield as a function of blast efficiency factor and yield scaling factor

Compute scale factors for impulse, pressure, time and distance via SCALE routine (scale factors stored in SCALT common)

Compute maximum potential damage limit, burst lifetime and fireball radius at time of burst

Compute source levels for radiation fluence factors – prompt gamma, neutron and thermal via NUCSLEV routine (factors stored in NWCHAR common)

Loop to compute environment intensities and damage levels for all Red units

N: WRITE ERROR MESSAGE

at time of burst

Compute source levels for radiation fluence factors – prompt gamma, neutron and thermal via NUCSLEV routine (factors stored in NWCHAR common)

Loop to compute environment intensities and damage levels for all Red units

Do for all Red units

Red unit in game?

Y

Store functional type of Red unit

Red unit still alive?

Y

Compute Red unit's distance from burst

Red unit closer than four times potential damage limit?

Y

Red unit within fireball?

Y

SET DAMAGE ARRAYS, VARIABLES AND FLAGS

N

COMPUTE RED UNIT CURRENT VELOCITY

RED UNIT WITHIN POTENTIAL DAMAGE LIMIT OR VELOCITY BRINGING IT TO MEET SHOCK FRONT?

Y

INITIALIZE ENVIRONMENT ARRAYS

COMPUTE SCALE FACTORS FOR TARGET ALTITUDE VIA SCALE ROUTINE

COMPUTE TIME TARGET WILL INTERCEPT SHOCK FRONT VIA MULLER ROUTINE

COMPUTE SLANT RANGE FROM BURST TO SHOCK INTERCEPT VIA RSHK FUNCTION

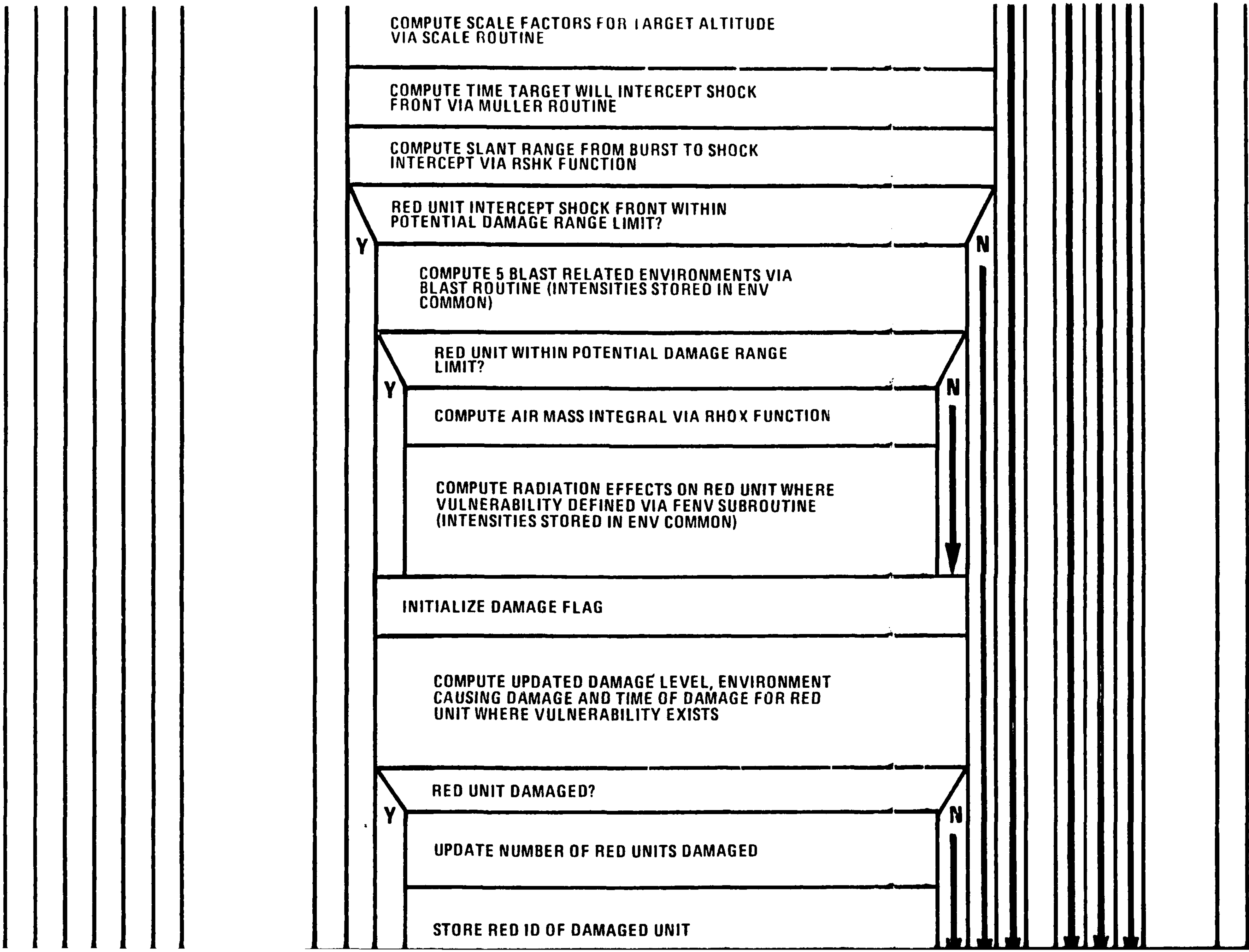
RED UNIT INTERCEPT SHOCK FRONT WITHIN POTENTIAL DAMAGE RANGE LIMIT?

N

N

N

N


COMPUTE SCALE FACTORS FOR TARGET ALTITUDE VIA SCALE ROUTINE
COMPUTE TIME TARGET WILL INTERCEPT SHOCK FRONT VIA MULLER ROUTINE
COMPUTE SLANT RANGE FROM BURST TO SHOCK INTERCEPT VIA RSHK FUNCTION
RED UNIT INTERCEPT SHOCK FRONT WITHIN POTENTIAL DAMAGE RANGE LIMIT?
Y
N
COMPUTE 5 BLAST RELATED ENVIRONMENTS VIA BLAST ROUTINE (INTENSITIES STORED IN ENV COMMON)
RED UNIT WITHIN POTENTIAL DAMAGE RANGE LIMIT?
Y
N
COMPUTE AIR MASS INTEGRAL VIA RHOX FUNCTION
COMPUTE RADIATION EFFECTS ON RED UNIT WHERE VULNERABILITY DEFINED VIA FENV SUBROUTINE (INTENSITIES STORED IN ENV COMMON)
INITIALIZE DAMAGE FLAG
COMPUTE UPDATED DAMAGE LEVEL, ENVIRONMENT CAUSING DAMAGE AND TIME OF DAMAGE FOR RED UNIT WHERE VULNERABILITY EXISTS
RED UNIT DAMAGED?
Y
N
UPDATE NUMBER OF RED UNITS DAMAGED
STORE RED ID OF DAMAGED UNIT

3

COMPUTE UPDATED DAMAGE LEVEL, ENVIRONMENT CAUSING DAMAGE AND TIME OF DAMAGE FOR RED UNIT WHERE VULNERABILITY EXISTS

RED UNIT DAMAGED?

Y

UPDATE NUMBER OF RED UNITS DAMAGED

STORE RED ID OF DAMAGED UNIT

N

Loop to compute environment intensities and damage levels for all Blue ships

Do for all Blue ships

Store functional type of Blue ship

Blue ship still alive?

Y

Compute Blue ship's distance from the burst

Blue ship within potential damage range limit?

Y

Blue ship within fireball?

Y

VIA RSHK FUNCTION

UNIT POSSIBLE DAMAGE BY BLAST?

Y

BLUE AIRCRAFT INTERCEPT SHOCK FRONT WITHIN POTENTIAL DAMAGE RANGE?

Y

COMPUTE 5 BLAST RELATED ENVIRONMENTS VIA [illegible]

N

N

N

N

N

N

Blue ship within potential damage range limit?

Y / N

Blue ship within fireball?

Y / N

SET DAMAGE ARRAYS, VARIABLES AND FLAGS

INITIALIZE ENVIRONMENT ARRAYS

COMPUTE SCALE FACTORS FOR TARGET ALTITUDE VIA SCALE ROUTINE

COMPUTE TIME TARGET WILL INTERCEPT SHOCK FRONT VIA TSHK FUNCTION (TIME STORED IN ENV COMMON)

COMPUTE 5 BLAST RELATED ENVIRONMENTS VIA BLAST ROUTINE (INTENSITIES STORED IN ENV COMMON)

COMPUTE AIR MASS INTEGRAL VIA RHOX FUNCTION

COMPUTE RADIATION EFFECTS ON BLUE SHIP WHERE VULNERABILITY DEFINED VIA FENV SUBROUTINE (INTENSITIES STORED IN ENV COMMON)

INITIALIZE DAMAGE FLAG

COMPUTE UPDATED DAMAGE LEVEL, ENVIRONMENT CAUSING DAMAGE AND TIME OF DAMAGE FOR BLUE SHIP WHERE VULNERABILITY EXISTS

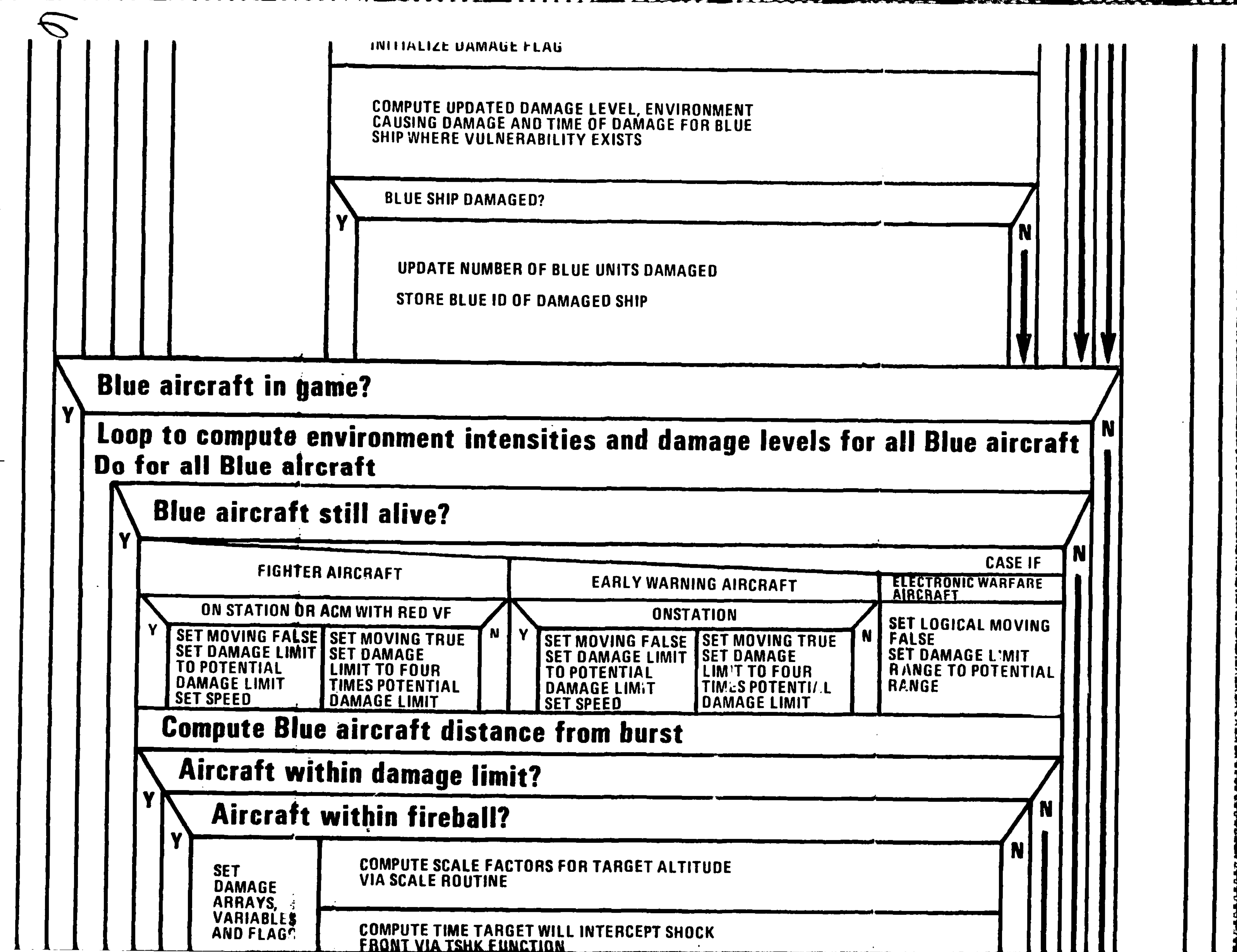
BLUE SHIP DAMAGED?

Y / N

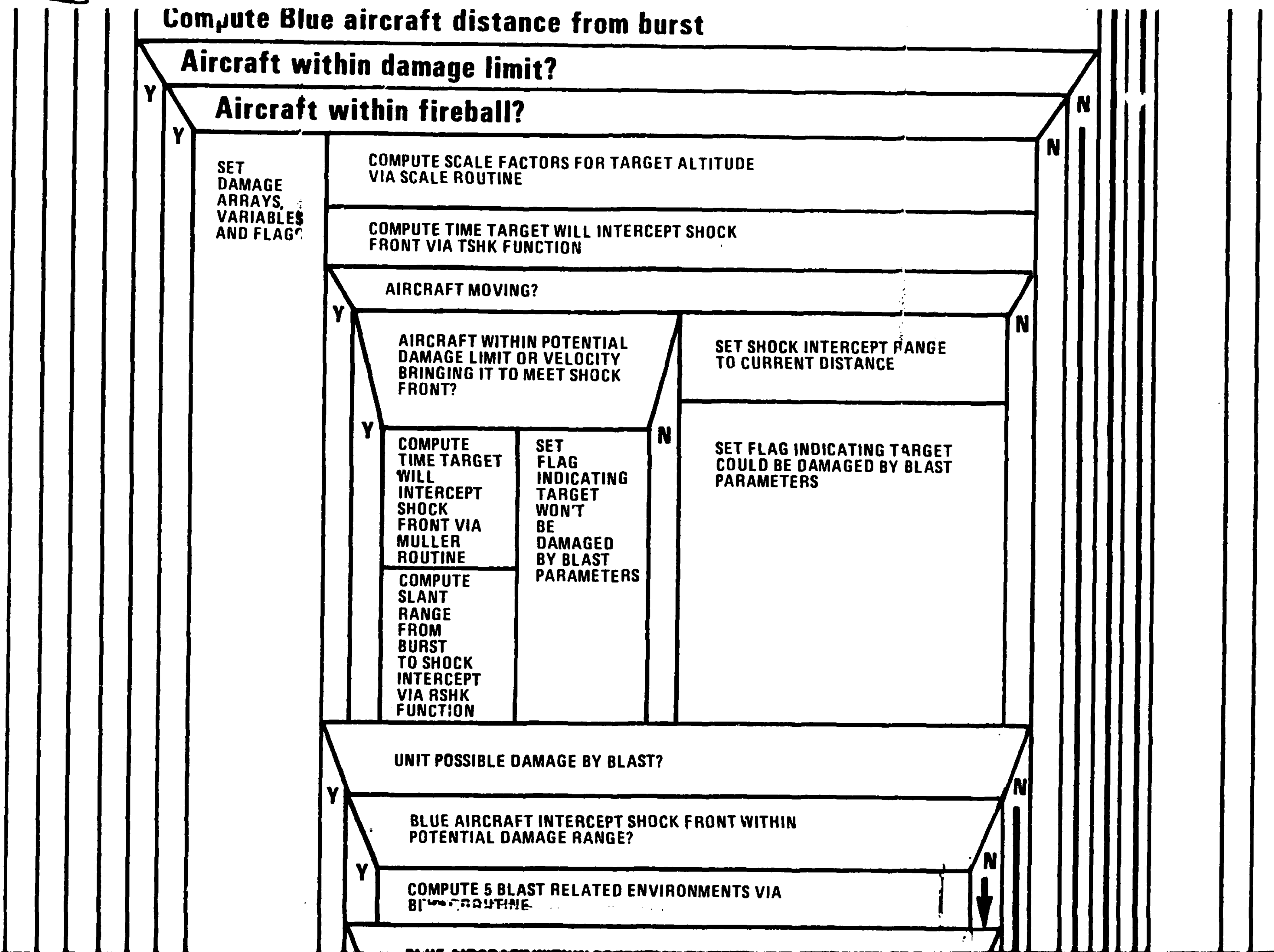
UPDATE NUMBER OF BLUE UNITS DAMAGED

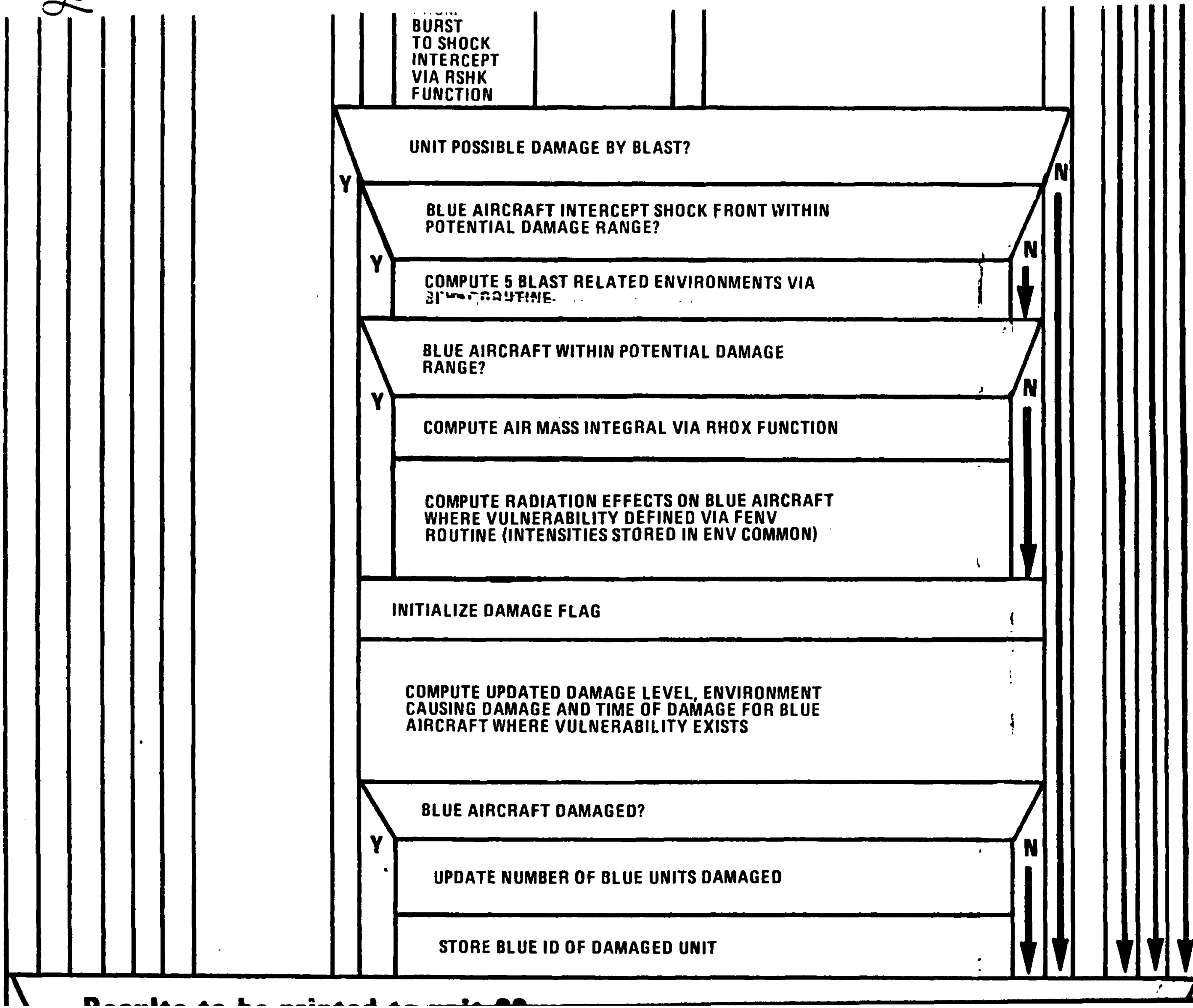
STORE BLUE ID OF DAMAGED SHIP

Blue aircraft in game?



6
INITIALIZE DAMAGE FLAG
COMPUTE UPDATED DAMAGE LEVEL, ENVIRONMENT CAUSING DAMAGE AND TIME OF DAMAGE FOR BLUE SHIP WHERE VULNERABILITY EXISTS
BLUE SHIP DAMAGED?
Y
N
UPDATE NUMBER OF BLUE UNITS DAMAGED
STORE BLUE ID OF DAMAGED SHIP
Blue aircraft in game?
Y
N
Loop to compute environment intensities and damage levels for all Blue aircraft
Do for all Blue aircraft
Blue aircraft still alive?
Y
N
CASE IF
FIGHTER AIRCRAFT
EARLY WARNING AIRCRAFT
ELECTRONIC WARFARE AIRCRAFT
ON STATION OR ACM WITH RED VF
Y
N
SET MOVING FALSE SET DAMAGE LIMIT TO POTENTIAL DAMAGE LIMIT SET SPEED
SET MOVING TRUE SET DAMAGE LIMIT TO FOUR TIMES POTENTIAL DAMAGE LIMIT
ONSTATION
Y
N
SET MOVING FALSE SET DAMAGE LIMIT TO POTENTIAL DAMAGE LIMIT SET SPEED
SET MOVING TRUE SET DAMAGE LIMIT TO FOUR TIMES POTENTIAL DAMAGE LIMIT
SET LOGICAL MOVING FALSE SET DAMAGE LIMIT RANGE TO POTENTIAL RANGE
Compute Blue aircraft distance from burst
Aircraft within damage limit?
Y
N
Aircraft within fireball?
Y
N
SET DAMAGE ARRAYS, VARIABLES AND FLAGS
COMPUTE SCALE FACTORS FOR TARGET ALTITUDE VIA SCALE ROUTINE
COMPUTE TIME TARGET WILL INTERCEPT SHOCK FRONT VIA TSHK FUNCTION


Compute Blue aircraft distance from burst
Aircraft within damage limit?
Y
N
Aircraft within fireball?
Y
N
SET DAMAGE ARRAYS, VARIABLES AND FLAGS
COMPUTE SCALE FACTORS FOR TARGET ALTITUDE VIA SCALE ROUTINE
COMPUTE TIME TARGET WILL INTERCEPT SHOCK FRONT VIA TSHK FUNCTION
AIRCRAFT MOVING?
Y
N
AIRCRAFT WITHIN POTENTIAL DAMAGE LIMIT OR VELOCITY BRINGING IT TO MEET SHOCK FRONT?
Y
N
COMPUTE TIME TARGET WILL INTERCEPT SHOCK FRONT VIA MULLER ROUTINE
COMPUTE SLANT RANGE FROM BURST TO SHOCK INTERCEPT VIA RSHK FUNCTION
SET FLAG INDICATING TARGET WON'T BE DAMAGED BY BLAST PARAMETERS
SET SHOCK INTERCEPT RANGE TO CURRENT DISTANCE
SET FLAG INDICATING TARGET COULD BE DAMAGED BY BLAST PARAMETERS
UNIT POSSIBLE DAMAGE BY BLAST?
Y
N
BLUE AIRCRAFT INTERCEPT SHOCK FRONT WITHIN POTENTIAL DAMAGE RANGE?
Y
N
COMPUTE 5 BLAST RELATED ENVIRONMENTS VIA


BURST
TO SHOCK
INTERCEPT
VIA RSHK
FUNCTION
UNIT POSSIBLE DAMAGE BY BLAST?
Y
N
BLUE AIRCRAFT INTERCEPT SHOCK FRONT WITHIN POTENTIAL DAMAGE RANGE?
Y
N
COMPUTE 5 BLAST RELATED ENVIRONMENTS VIA
BLUE AIRCRAFT WITHIN POTENTIAL DAMAGE RANGE?
Y
N
COMPUTE AIR MASS INTEGRAL VIA RHOX FUNCTION
COMPUTE RADIATION EFFECTS ON BLUE AIRCRAFT WHERE VULNERABILITY DEFINED VIA FENV ROUTINE (INTENSITIES STORED IN ENV COMMON)
INITIALIZE DAMAGE FLAG
COMPUTE UPDATED DAMAGE LEVEL, ENVIRONMENT CAUSING DAMAGE AND TIME OF DAMAGE FOR BLUE AIRCRAFT WHERE VULNERABILITY EXISTS
BLUE AIRCRAFT DAMAGED?
Y
N
UPDATE NUMBER OF BLUE UNITS DAMAGED
STORE BLUE ID OF DAMAGED UNIT

9

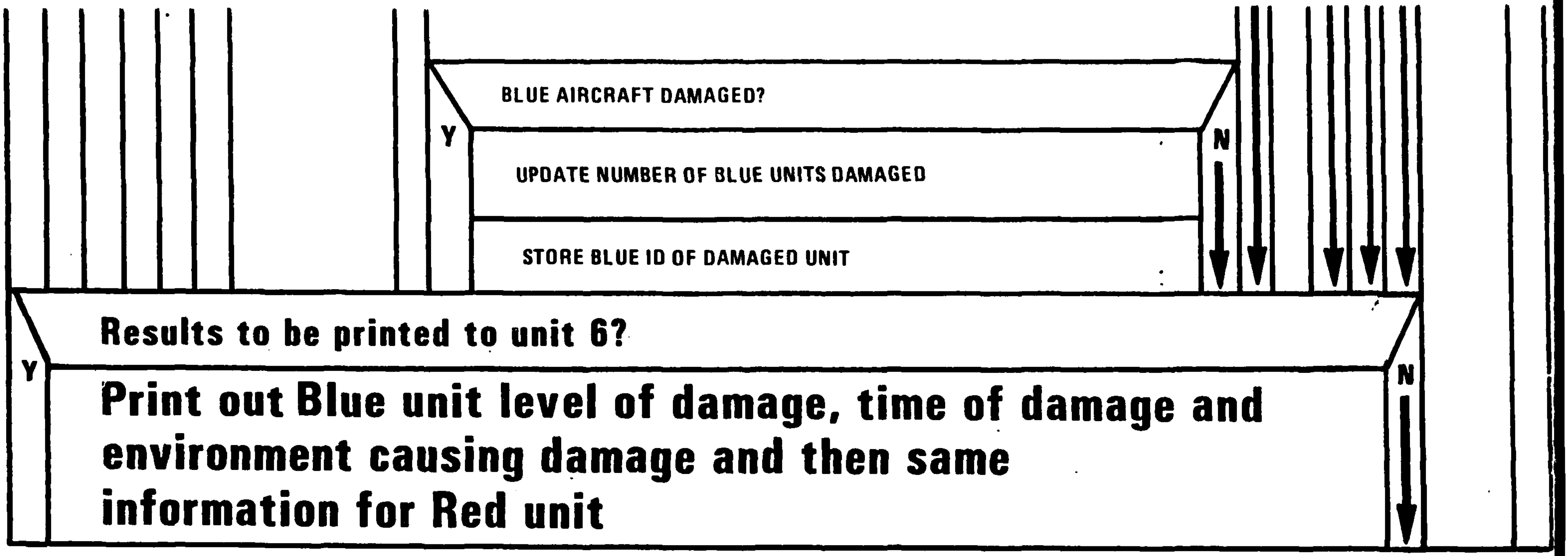


Figure 11-6 (U)

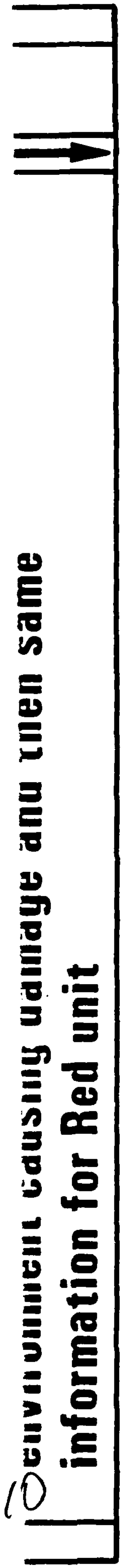


Figure 11-6 (U) Flow Diagram for Nucler (U)

The effective blast yield (Yeff) is then computed as a function of blast efficiency factor, yield and burst altitude (Fig. 2-64, EM-1):

$$Yeff = Y \times e^{A \times Zb + B \times Zb^2}, \text{ where}$$

Y - yield (kt),
Zb - burst altitude (m),
A = $-9.1788423 \times 10^{-6}$, and
B = $-2.67170618 \times 10^{-10}$.

Yield scale factor (Ys) is then computed as:

$$Ys = Yeff^{1/3}, \text{ where}$$

Yeff - effective yield (kt).

Modified Sachs scale factors for impulse (Si), pressure (Sp), time (St) and distance (Sd) are then computed via SCALE routine. A maximum range for potential damage (RLIMIT) is then computed via the following equation:

$$RLIMIT = 3048 \times Ys \times Sd, \text{ where}$$

Ys - yield scale factor, and
Sd - distance scale factor.

The maximum time for damage to occur (Tmax) is also computed as:

$$TMAX = 8.6 \times Ys \times St, \text{ where}$$

Ys - yield scale factor
St - time scale factor

The fireball radius at burst time (RO) is also computed via:

$$RO = 34 \times 2.16^{\log(Yeff)} \times 2^{Zb/20000} \text{, where}$$

Yeff - effective yield (kt), and
Zb - burst altitude (m).

The potential damage range and the fireball radius are used as the basis for excluding unnecessary computations. Next, the source levels for radiation fluence factors are computed via a call to NUCSLEV. Every Red unit and every Blue unit is considered as a potential victim for each of the nuclear environments. The full list of potential victims is divided into three major groups due to different array names and dimensions that must be referenced. The first group is the Red units, the second group the Blue ships and the third group the Blue aircraft.

The same general pattern of computations is used in each of the three groups. A succession of tests is applied to screen out as many units as feasible from more detailed computations. Each unit is eliminated from further consideration by the following tests:

1. It has already left the game.

2. It is more than four times the potential damage range limit and thus could not intercept the shock front within this range limit.

3. It is within the fireball so the unit is destroyed.

4. It is beyond the potential damage range limit and its velocity is not bringing it closer fast enough to meet the shock front within the range limit.

5. It intercepts the shock front beyond the potential range damage limit and thus will not be subject to blast effects.

6. If it is beyond the potential damage range limit at burst time it will not be subject to radiation effects, regardless of its velocity.

If a potential victim cannot be eliminated by the above tests, then the appropriate subroutines are called to compute the intensity of each of the environments for which the platform type has a vulnerability criterion specified in the XXRSP arrays. The computed intensity of each of the environments for each unit is then compared with the criteria for each damage level (K), and the maximum damage level for each unit is recorded along with the particular environment (J) that inflicted the maximum damage and the time the damage occurred.

When NUCLER has performed the above procedure for all environments against all victims, it returns to the calling subroutine.

11.3.1 Subprograms required

The following list of FORTRAN subroutines is required to support the operation of NUCLER:

BLAST	Computes peak overpressure, dynamic overpressure, impulse and particle velocity. The routine returns the five blast-related environment intensities and the time of each environment.
BLINT	Computes the time and position of the intercept between the shock front and a moving target.
FENV	Computes the radiation intensities - prompt gamma peak dose rate, neutron fluence and thermal fluence.
MATM62	A standard atmosphere model that returns pressure, temperature, sound speed and density as a function of altitude.
MULLER	Computes any described number of zeros, real or complex, of an arbitrary function using MULLER's method.
NUCSLEV	Computes the source levels for prompt gamma, neutron fluence and thermal fluence as a function of burst altitude.
SCALE	Computes scaling factors for distance, time, pressure and impulse to permit fitting to standard curves for 1 kiloton sea level data.
VPARTS	Resolves the particle velocity into axial and transverse components for moving targets.

In addition to the foregoing subroutines, NUCLER also requires the use of the following FORTRAN function subprograms:

I4TO2	Converts INTEGER*4 to INTEGER*2.
MYSTEM	Computes the height of the Mach Y stem (triple point altitude) at the scaled horizontal range.
PFA	Computes peak overpressure (psi) versus slant range for a 1 kiloton burst in free air.
PMR	Computes peak overpressure (psi) in the Mach region at sea level based on DASA 1200 data with the assumption of essentially plane ground surface and negligible thermal effects.
PULSE	Computes overpressure impulse for low altitude targets.
RHOX	Computes the air mass integral between two points above a flat earth.
RSHK	Computes slant range of shock wave for a free air burst in a sea level environment as a function of time.
TERPL	Performs a linear interpolation.
TERPL2	Performs a linear interpolation of matrix or grid data.
TSHK	Computes shock front arrival time as a function of yield, range and altitude.
XYDIST	Computes planar distance between two points.
XYZDST	Computes three-dimensional distance between two points.

11.3.1.1 Subroutine BLAST

Subroutine BLAST is called by NUCLER to compute the blast effects due to a nuclear burst. The routine computes peak overpressure and dynamic pressure for all types of targets, overpressure impulse for ship targets only and shock

particle velocity across and along the flight path for aircraft targets.

The subroutine first computes the scaled height of burst and the scaled target range. If the scaled height of burst is greater than or equal to 609.6 meters or 2,000 feet, the burst will not create an incidence wave so peak overpressure (P1) is computed using the PFA function. This function requires as input the distance from a 1 kiloton burst (D1) which is obtained from the following equation:

$$D1 = D / (Ys \times Sd) \text{ , where}$$

D - distance of target from burst (m),
Ys - scaled yield, and
Sd - distance scale factor.

The peak overpressure at the target is then computed as:

$$P = P1 \times Sp \text{ , where}$$

P - peak overpressure at target for current burst (psi),
P1 - peak overpressure at target for 1 kt burst (psi), and
Sp - overpressure scale factor.

Next, a check is performed to see if the target is a ship whose scaled range is less than or equal to the scaled height of burst. If it is, then peak overpressure is compensated for a normally incident shock wave (Pc) via the following equation:

$$Pc = 2 \times P \times ((1 + (4P / Pa)) / (1 + (P / Pa))),$$

where

P - peak overpressure at target for current burst (psi), and
Pa - pressure at target altitude (psi).

If the scaled height of burst is between 48.77 meters and 609.6 meters, then the target position is checked to see if it is within the Mach Y stem. If it is then the peak overpressure is computed via the PMR function. Again a check is done to determine if the target is within the incidence wave region. If it is, the peak overpressure is compensated for as above. If the target is not within the Mach Y

stem, the peak overpressure is computed using the PFA function as above. Again a check is done to determine if the target is a ship in the normally incidence wave region and if it is then a compensation occurs. Finally, if the burst is less than 48.77 meters then it is assumed to be a ground burst and an additional concentration factor of 1.2 is used with the PFA function to compute peak overpressure.

Now the effects on the target are computed for those targets having vulnerability arrays defined. The dynamic pressure is computed via an EM-1, Table A.1 equation for an ideal gas with a gamma of 1.4:

$$Pd = 2.5 \times P / (7 \times Pa + P)^{2} \quad , \text{where}$$

P - peak overpressure (P) at target or compensated peak overpressure at target (Pc) in psi, and
Pa - pressure at target altitude (psi).

If the target is a ship with vulnerability criterion defined and the scaled height of burst lies between 500 and 800 meters, then the overpressure impulse is computed via the PULSE function. If the target is an aircraft with vulnerability criterion defined, the shock particle velocity is computed as:

$$V = Ca / ((1 + (6 \times P) / (7 \times Pa))^{1/2}) \quad ,$$

where

Ca - speed of sound at target altitude (m/sec),
P - peak overpressure at target or compensated peak overpressure at target, Pc, (psi), and
Pa - pressure at target altitude (psi).

The along axis and across axis components of particle velocity are then computed via VPARTS subroutine. All of the applicable effects and the time of that effect are stored for future use. The subroutine then returns to NUCLER.

A flow diagram for BLAST is included as Figure 11-7.

11.3.1.2 Subroutine BLINT

This subroutine computes the distance separating a nuclear explosion generated shock wave and a moving target position. The routine is called by MULLER to calculate the point of intercept of a target and the burst generated

BLAST

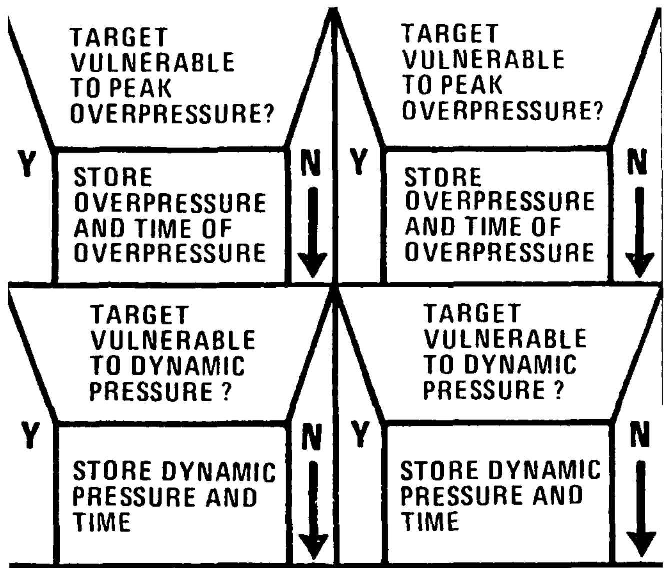
Set flag for type of target

Compute speed of sound and atmospheric pressure at target altitude via MATM62 routine

Compute scaled height of burst and scaled range of target from burst

Case if

- SCALED HEIGHT OF BURST ABOVE MACH Y STEM
  - SET MACH FLAG TO FALSE
  - COMPUTE PEAK OVERPRESSURE VIA PFA FUNCTION
  - IS TARGET A SHIP WHOSE RANGE IS LESS THAN OR EQUAL TO SCALED HEIGHT OF BURST ?
    - Y: COMPENSATE FOR NORMALLY INCIDENT BLAST WAVE
    - N
- SCALED HEIGHT OF BURST WITHIN MACH Y STEM
  - TARGET WITHIN MACH Y STEM ?
    - Y:
      - SET MACH FLAG TO TRUE
      - COMPUTE PEAK OVER-PRESSURE VIA PMR FUNCTION
      - RESET MACH FLAG IN PMR
      - IS TARGET OUT OF MACH REGION AND RANGE WITHIN SCALED HEIGHT OF BURST ?
    - N:
      - SET MACH FLAG TO FALSE
      - COMPUTE PEAK OVER-PRESSURE VIA PFA FUNCTION
      - IS TARGET A SHIP WITHIN RANGE OF SCALED HEIGHT OF BURST ?
        - Y: COMPENSATE FOR
        - N
- SCALED HEIGHT OF BURST BELOW MACH Y STEM
  - TREAT BURST AS GROUND BURST AND COMPUTE PEAK OVERPRESSURE WITH CONCENTRATION FACTOR OF 1.2
  - SET MACH FLAG TO FALSE


TARGET VULNERABLE TO PEAK OVERPRESSURE?
Y
STORE OVERPRESSURE AND TIME OF OVERPRESSURE
N
TARGET VULNERABLE TO PEAK OVERPRESSURE?
Y
STORE OVERPRESSURE AND TIME OF OVERPRESSURE
N
TARGET VULNERABLE TO DYNAMIC PRESSURE ?
Y
STORE DYNAMIC PRESSURE AND TIME
N
TARGET VULNERABLE TO DYNAMIC PRESSURE ?
Y
STORE DYNAMIC PRESSURE AND TIME
N

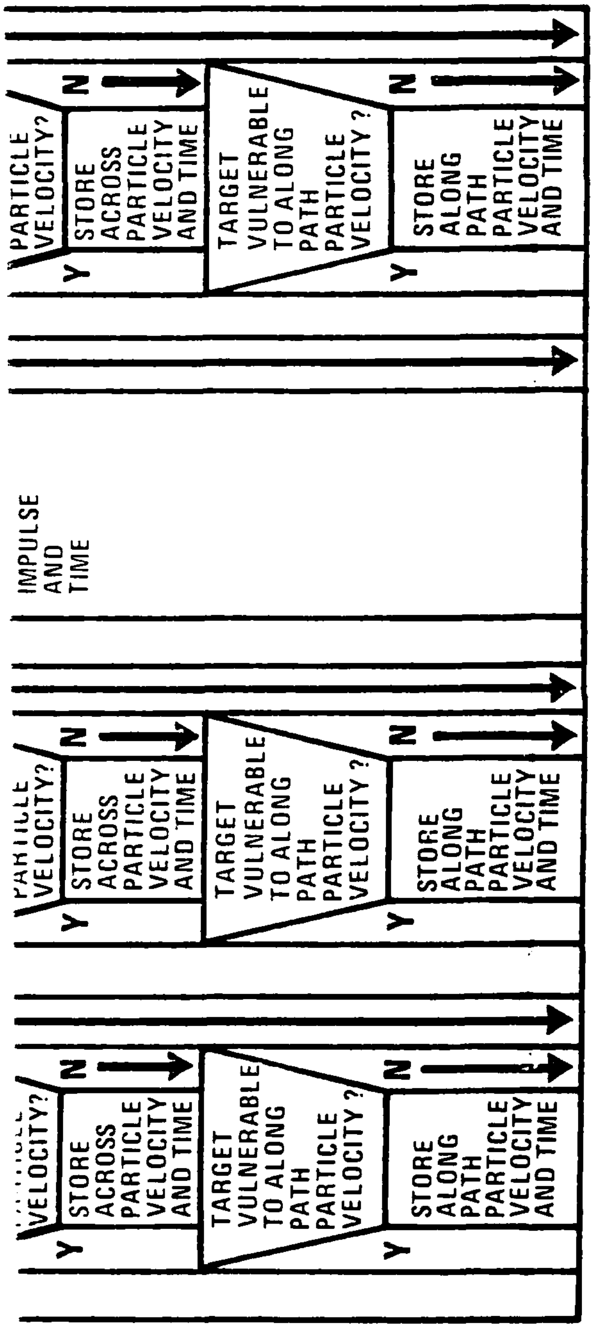


Figure 11-7 (U) Subroutine Blast (U)

3

shock front.

The routine first computes the updated distance from the burst to the target and then computes the distance from the burst to the shock front utilizing the RSHK function. Then these two distances are used to compute the distance from the target to the shock front.

11.3.1.3 Subroutine FENV

This subroutine is called by NUCLER to compute the radiation intensity for peak gamma dose rate, neutron fluence and thermal fluence if vulnerability criterion are defined for the current target using mass integral scaling.

For computing the peak gamma dose rate and neutron fluence, the routine first computes a buildup factor via the TERPL function from EM-1, Figure 5-7, Page 5-15. A buildup factor is used to compensate for the scattered neutrons and gamma rays reaching the target position from other directions. The TERPL function interpolates between a given mass integral scaling number and the natural log of a given fraction of either peak gamma dose rate or vacuum neutron fluence for a given mass integral scaling number. The peak gamma dose rate (GAMDOSE) and neutron fluence (NFLU) are computed as a function of source level, build up factor and distance of target from burst via the following equations:

$$GAMDOSE = SG \times B / (4 \times PI \times R^2) \text{ and}$$

$$NUFLU = SN \times B / (4 \times PI \times R^2) \text{, where}$$

B - build up factors,
PI = 3.1415927,
R - distance from burst to target (cm),
SG - prompt gamma source level (rad(si)/sec), and
SN - neutron fluence source level (neutrons).

The thermal fluence (Q) is computed as a function of source level and distance of target from burst via the following equation:

$$Q = ST / (4 \times PI \times R^2) \text{ , where}$$

ST - effective thermal source level (cal),
PI = 3.1415927, and
R - distance from burst to target (cm).

These environment intensities and the times the intensities occurred are stored in ENV common for future use. The routine then returns to the calling routine NUCLER.

A flow diagram for FENV is included as Figure 11-8.

11.3.1.4 Subroutine NUCSLEV

This subroutine is called by NUCLER to compute the radiation source levels of peak gamma dose rate, neutron fluence and thermal fluence as a function of burst altitude. The routine first obtains the thermal partition value for this particular thermal effective yield and burst altitude via the TERPL2 function utilizing data from EM-1, page 3.4, Figure 3-1. Once the correct thermal partition value is obtained, the effective thermal source level (ST) for this burst is calculated via the following equation in EM-1 section 3-17:

$$ST = THP \times Y \times K \text{ , where}$$

THP - thermal partition value for this yield,
Y - yield (kt), and
K = 10^{12} (converts kiloton to calories).

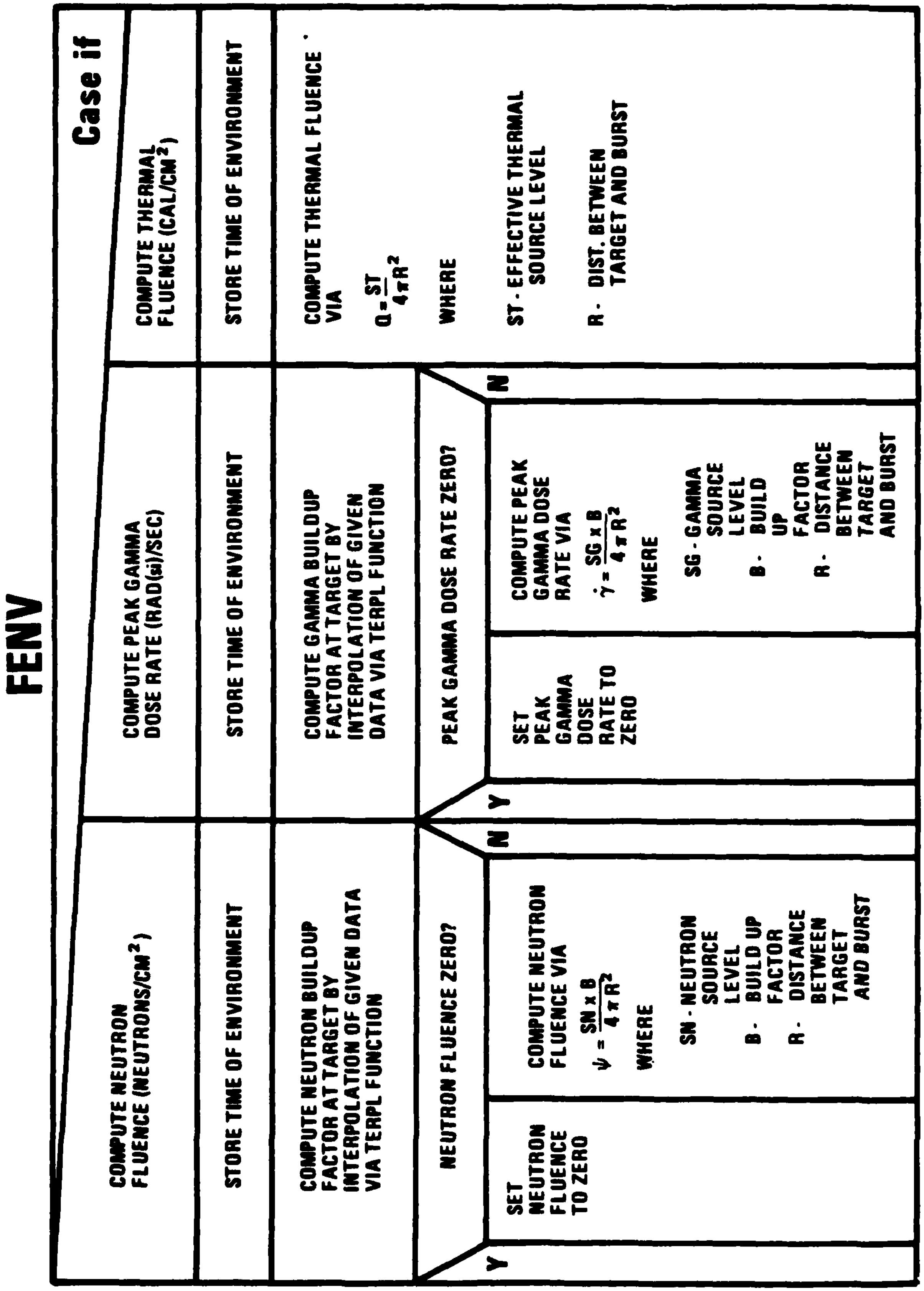


Figure 11-8 (U) Flow Diagram for FENV (U)

Next, if the height of burst is less than 300 meters, the neutron source level (SN) is corrected for burst altitude via a curve fit to the curve in EM-1, page 5-37 utilizing the following equation:

$$SN = NO \times B \times e^{A \times Zb + C} \quad \text{, where}$$

NO - total number of neutrons emitted for this warhead type,
A = 1.6667,
B = 1 / 6.2483,
C = 17.126, and
Zb - burst altitude (m).

Finally, the source level for this burst is compensated for burst altitude if the burst is a surface burst; i.e., the burst altitude is less than or equal to the fireball radius at time of burst via the following equation:

$$SG = 2 / 3 \times SG1 \quad \text{, where}$$

$$SG1 = Y \times FG \times K \times GDOSE / GAMDT \quad \text{,where}$$

Y - yield (kt),
FG - prompt gamma energy fraction (dimensionless),
K = 10^{12} (converts kiloton to calories),
GDOSE - gamma fluence to dose conversion factor (Rad(si) - cm/cal), and
GAMDT - effective prompt gamma pulse width (seconds).

11.3.1.5 Subroutine SCALE

This routine is called by NUCLER to compute the modified SACHS scaling factors for a 1 kiloton burst. The routine first compensates for high altitudes above near surface bursts by using a fraction of the shock altitude. The routine then computes the pressure, speed of sound, density and temperature at the given altitude via MATM62 routine. Then the scale factors for pressure (Sp), distance (Sd), time (St) and impulse (Si) are computed as in EM-1, page 2-23, via the following equations:

$$Sp = Pa / P0 ,$$

$$Sd = 1 / Sp^{1/3} ,$$

$$St = Sd \times C0/Ca ,$$

$$Si = St \times Sp , \text{ where}$$

Pa - atmospheric pressure at given altitude,
P0 - pressure at sea level,
Ca - speed of sound at given altitude, and
C0 - speed of sound at sea level.

11.3.1.6 Subroutine VPARTS

This routine is called by NUCLER to resolve the particle velocities into axial and transverse components relative to the flight path of a moving target. The routine first computes the target's speed and if the target has a velocity, the velocity is updated to the time of shock and a new target position is computed. Then the component velocities are computed via the following equations:

$$Va = V \times \cos(\text{Theta}) \text{ and}$$
$$Vc = V \times \sin(\text{Theta}) , \text{ where}$$

V - particle velocity (m/sec)
Theta - angle between the target velocity vector and relative position vector

A flow diagram for VPARTS is included as Figure 11-9.

11.3.1.7 Function MYSTEM

This logical function is called by BLAST to calculate the height of the Mach Y stem at the scaled horizontal range. If MYSTEM is true, the target is in the reflection region otherwise MYSTEM is false. If the scaled range is less than 42.67 meters, MYSTEM is set to false. If the scaled range is greater than or equal to 42.67 meters, the coefficients to be used in the triple point height estimating function are computed and the triple point is then computed. Now, if the triple point height is greater than or equal to the scaled target height then MYSTEM is true, otherwise it is false.

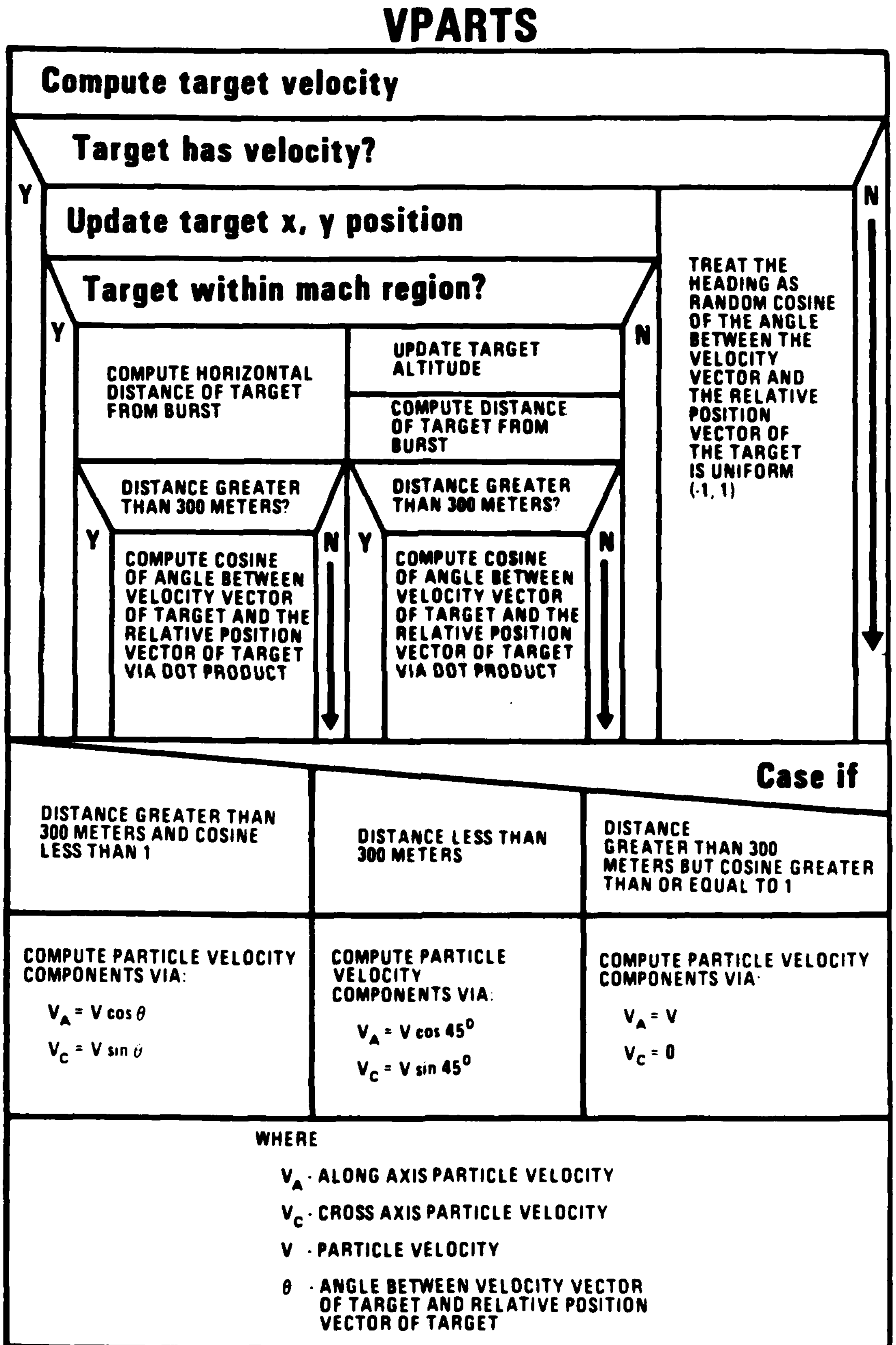


Figure 11-9 (U) Flow Diagram for VPARTS (U)

11.3.1.8 Function PFA

This function is used to compute the peak overpressure for a 1 kiloton burst at a specified slant range. The function performs a numerical fit to AFWL 1 kiloton curve in EM-1, page 2-7, Figure 2-2.

11.3.1.9 Function PMR

This function is used to compute peak overpressure in the Mach region at sea level using DASA 1200 data with the assumption of essentially plane ground surface and negligible thermal effects. First, the function computes the three heights of burst: zero, an intermediate height of burst and the height of burst for which the ground point of interest is just within the Mach region. The data for the three heights of burst are then used to calculate linear interpolation factors for the overpressure data. Then the overpressure is computed from either the burst data or the PFA function.

11.3.1.10 Function PULSE

This function calculates overpressure impulse for targets below 60 meters. The impulse is calculated using a DNA EM-1 database for near ideal surface. PULSE interpolates the given database utilizing the TERPL2 function.

11.3.1.11 Function RHOX

This function determines the air mass integral between two points above a flat earth. If the two points are not coaltitudinal, the air mass integral is computed for both altitudes via a table interpolation using the TERPL function. Then the air mass integral between the two points is calculated using equation from EM-1 page 5-12:

$$RHOX = ((R2 - R1) \times D) / (Z2 - Z1)$$, where

R1 - air mass integral of point one (gm/cm^2),
R2 - air mass integral of point two (gm/cm^2),
D - distance between the two points (m),
Z1 - altitude of point one (m), and
Z2 - altitude of point two (m).

If the two points are coaltitudinal, then the MATM62 routine is used for the air mass integral calculation.

11.3.1.12 Function RSHK

This function computes the slant range of a shock wave for a free air burst in a sea level environment at a given time after burst. The routine interpolates the AFWT 1 kiloton standard curve of scaled range versus scaled time, Figure 1.1-6, DNA 2048-1, "Handbook for Analysis of Nuclear Weapon Effects on Aircraft". If the time after burst is less than 0.0, the slant range of shock wave is set to zero. In addition, if the time after burst is out of range of the data; i.e., greater than 8.3 seconds, the slant range is set to a maximum value.

11.3.1.13 Function TSHK

This function computes the time the shock wave reaches a given slant range for a free air burst in a sea level environment. The routine interpolates the AFWT 1 kiloton standard curve of scaled range versus scaled time, Figure 1.1-6, DNA 2048-1, "Handbook for Analysis of Nuclear Weapon Effects on Aircraft". If the scaled range is negative, the time of the shock wave is set to zero and if the scaled range is beyond the range of the data, i.e. greater than 3048 meters, the time of the shock is set to a maximum value.

END

1-87

DTIC